2020-12

IMPROVING OPERATIONAL REPORTING WITH ARTIFICIAL INTELLIGENCE

Bailey, George W.

Monterey, CA; Naval Postgraduate School

http://hdl.handle.net/10945/66578

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

IMPROVING OPERATIONAL REPORTING WITH ARTIFICIAL INTELLIGENCE

by

George W. Bailey

December 2020

Thesis Advisor:	Timothy C. Warren
Co-Advisor:	Arijit Das

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC, 20503.			
1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** December 2020	**3. REPORT TYPE AND DATES COVERED** Master's thesis	
4. TITLE AND SUBTITLE IMPROVING OPERATIONAL REPORTING WITH ARTIFICIAL INTELLIGENCE			**5. FUNDING NUMBERS**
6. AUTHOR(S) George W. Bailey			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.			**12b. DISTRIBUTION CODE** A

13. ABSTRACT (maximum 200 words)

Today, military analysts receive far more information than they can process in the time available for mission planning or decision-making. Operational demands have outpaced the analytical capacity of the Department of Defense. To address this problem, this work applies natural language processing to cluster reports based on the topics they contain, provides automatic text summarizations, and then demonstrates a prototype of a system that uses graph theory to visualize the results. The major findings reveal that the cosine similarity algorithm applied to vector-based models of documents produced statistically significant predictions of document similarity; the Term Frequency-Inverse Document Frequency algorithm improved similarity algorithm performance and produced topic models as document summaries; and a high degree of analytic efficiency was achieved using visualizations based on centrality measures and graph theory. From these results, one can see that clustering reports based on semantic similarity offers substantial advantages over current analytical procedures, which rely on manual reading of individual reports. On this basis, this thesis provides a prototype of a system to improve the utility of operational reporting as well as an analytical framework that can assist in the development of future capabilities for military planning and decision-making.

14. SUBJECT TERMS similarity, natural language processing, NLP, machine learning, artificial intelligence, term frequency, inverse document frequency, TF-IDF, China, graph, cosine, Euclidean, Jaccard			**15. NUMBER OF PAGES** 79
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

IMPROVING OPERATIONAL REPORTING WITH ARTIFICIAL INTELLIGENCE

George W. Bailey
Major, United States Army
BS, Murray State University, 2007

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN DEFENSE ANALYSIS (IRREGULAR WARFARE)

from the

NAVAL POSTGRADUATE SCHOOL
December 2020

Approved by: Timothy C. Warren
Advisor

Arijit Das
Co-Advisor

Douglas A. Borer
Chair, Department of Defense Analysis

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Today, military analysts receive far more information than they can process in the time available for mission planning or decision-making. Operational demands have outpaced the analytical capacity of the Department of Defense. To address this problem, this work applies natural language processing to cluster reports based on the topics they contain, provides automatic text summarizations, and then demonstrates a prototype of a system that uses graph theory to visualize the results. The major findings reveal that the cosine similarity algorithm applied to vector-based models of documents produced statistically significant predictions of document similarity; the Term Frequency-Inverse Document Frequency algorithm improved similarity algorithm performance and produced topic models as document summaries; and a high degree of analytic efficiency was achieved using visualizations based on centrality measures and graph theory. From these results, one can see that clustering reports based on semantic similarity offers substantial advantages over current analytical procedures, which rely on manual reading of individual reports. On this basis, this thesis provides a prototype of a system to improve the utility of operational reporting as well as an analytical framework that can assist in the development of future capabilities for military planning and decision-making.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

AI	artificial intelligence
AIC	Akaike information criterion
DFM	document feature matrix
DOD	Department of Defense
IDF	inverse document frequency
IIR	intelligence information report
JWICS	Joint Worldwide Intelligence Communications System
MGRS	military grid reference system
ML	machine learning
NLP	natural language processing
OPSUM	operations summary
R^n	Euclidean n-space
RMSE	root mean square error
SIPRNET	SECRET Internet Protocol Router Network
SITREP	situation report
SOF	special operations forces
TF	term frequency
TF-IDF	term frequency – inverse document frequency
URL	uniform resource locator

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

First and foremost, I am extremely grateful to my wife, Amanda, for her unwavering support and encouragement—she is the inspiration for everything that I do.

I would like to thank my thesis advisors, Dr. Timothy C. Warren and Research Associate Arijit Das, for their guidance, feedback, and support during this process.

I would like to thank Dr. John J. Arquilla for introducing me to systems thinking, which has served me well in my coursework, this thesis, and will throughout my life. I would like to thank Dr. Sean F. Everton for introducing me to dark networks and social network analysis, and Dr. Michael E. Freeman for introducing me to thinking critically about terrorist organizations. I would also like to thank Professor Carlos F. Borges for giving me a better understanding and greater appreciation for linear algebra and matrix theory. The amalgamation of these ideas and skills gave me the ability to complete this work.

I would also like to thank my professors and the Defense Analysis Department. This is a truly unique department that has not only expanded my knowledge of irregular warfare, but has provided me with an extremely valuable skillset that I would have been hard pressed to find in any other institution.

Most importantly, I thank God for giving me the opportunity and ability to understand and express these ideas.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

In 2017, the National Geospatial-Intelligence Agency's Director, Robert Cardillo, stated that a single sensor in one combat theatre generates daily the equivalent data of every game of three NFL seasons.[1] Processing all of this data is like analyzing the strategy behind every play, in every game, for three seasons, with three more coming tomorrow. That is just for a single sensor. The development of new information systems, collection platforms, and sensors has driven data collection. Today, military analysts receive far more information than they can process in the time available for mission planning or decision-making. Operational demands have outpaced the analytical capacity of the Department of Defense (DOD). Data that does not support decision-making is useless.

A. OPERATIONAL NEED

Data overload is a significant problem for Special Operations Forces (SOF). SOF units are generally smaller and have less manpower to analyze data than conventional forces. They also focus on a larger operational area per element, have high operational demands, and often plan for complex "grey zone" operations with ambiguous effectiveness measures. Additionally, the Special Operations community generates and relies on unstructured data, such as text, audio, images, and video. Much of this data is in the form of written narrative text such as Situation Reports (SITREPS), Operations Summaries (OPSUMs), or Intelligence Information Reports (IIRs). This type of data is the most difficult to store and analyze.

This is primarily a three-part problem—data analytics, and data storage, and retrieval. In terms of data analytics, SOF is not fully leveraging unstructured data. Operators spend a great deal of time writing reports that only a limited number of people can access or place in context. This is a significant opportunity loss. It is estimated that 95

[1] "GEOINT 2017 Symposium," National Geospatial-Intelligence Agency, June 6, 2017, https://www.nga.mil/news/1595643886627_GEOINT_2017_Symposium.html.

percent of all data produced within an organization is unstructured.[2] The unstructured nature of operational reports makes this information incredibly challenging to analyze. Each report requires a human reader for analysis. But the analysts spend a significant portion of their time sifting through a data container to find documents that contain relevant information. The second and third parts of this problem is that documents need to be organized for storage in a way that facilitates the retrieval. It is challenging to discover trends over time or aggregate information from multiple sources without a standardized storage method or central repository. This problem is magnified when the report creators are separated from the consumers. Reports sent to headquarters are not made available to adjacent units and are often lost during unit rotations. A large amount of this information is eventually forgotten and deleted because of upgrades in equipment (old drives become unusable with modern equipment), the context of the information is lost, or units move or need to make space for information related to current operations.

A potential solution to the problem is an area within Computer Science called Natural Language Processing (NLP). NLP algorithms improve narrative reporting's usefulness by reducing the workforce required to analyze and process reports for archiving. While reports such as SITREPS, OPSUMs, IIRs were the inspiration for this thesis, this research will apply to any report found in FM 101–5-2, U.S. *Army Report and Message Formats*, or any other written communication.

A primary motivation of this thesis is to better leverage internally produced data. The most reliable source of information that the DOD has access to is operational reporting. This data cannot be fully utilized given the current workforce. Automated methods such as NLP must be used to supplement the DOD's current analytical capacity. However, a lot of NLP research is focused on analyzing external data sources such as news sources for reporting trends, or social media for sentiment analysis. This is not an argument against using public sources of information; the process presented here can be used on this type of data. However, the DOD will gain a significant operational advantage from fully utilizing

[2] Amir Gandomi and Murtaza Haider, "Beyond the Hype: Big Data Concepts, Methods, and Analytics," *International Journal of Information Management* 35, no. 2 (April 1, 2015): 143, http://www.sciencedirect.com/science/article/pii/S0268401214001066.

internally produced data. Some of the most insightful sentiment on an issue should be derived from the reporting of Operators in the field, which will be reflected in their commentary.

B. RESEARCH QUESTIONS

The ultimate goal of this thesis is to advance research toward a system that can be used to join unstructured data in a way that assists mission planners. To do this, this thesis narrows its focus on specific aspects of unstructured data organization. First, a system that can group reports with similar topics; during mission planning, it is useful to have a technique of identifying SITREPs, OPSUMs, and IIRs containing details about the same subject. Second, a method of automatically extracting keywords to label documents based on their contents. Such a system of building intuitive labels would greatly increase productivity. Third, how to combine these metrics into a system that improves analytics. Advancing the knowledge on these three elements are foundational toward addressing the problems data analytics, data storage, and data retrieval.

This thesis will focus on these specific research questions.

1. How can NLP algorithms be used to measure similarity in reports?
2. How can NLP algorithms be used to automatically summarize reports?
3. How can the results of NLP algorithms be displayed in ways that facilitate decision-making?

C. CONCEPT OF DOCUMENT SIMILARITY

Before preceding further, it is important to define and briefly discuss the term “similarity” as it is used in this thesis. This concept is core to this research. In this thesis, similarity is referring to the likeness of the topics contained in a given set of reports. This is not referring to the report type, length, or format. Measuring similarity is the process of quantifying the topics so that its meaning can be compared to the meaning contained within other reports. Analysts and planners can then concentrate on a cluster of reports and spend less time reading through unrelated data.

D. RELATED RESEARCH

NLP is considered a subset of Artificial Intelligence (AI) and Machine Learning (ML). While the science of NLP is relatively evolved, it is considered one of the most significant challenges in AI. The algorithms in this thesis fall into a category of NLP commonly called statistical NLP. Much literature points out that the terminology is misleading, and differences between other forms of NLP is not clearly defined.[3] Statistical NLP is not limited to statistics, it includes all quantitative methods of automated language processing. Here statistical NLP algorithms are defined as those that rely on principles of linear algebra, probabilities, and graph theory. These methods were chosen intentionally because of the intended military application of this work. The NLP algorithms used here do not require large amounts of training data, the internal working of the algorithm is not proprietary and can easily be rebuilt on classified systems, and the algorithms are lightweight and can be run on a standard computer.

The basis for much of the work in NLP relies on the process of numerically quantifying reports in some way that can be measured.[4] A prevalent method of doing this is the vector space model. This is sometimes referred to as a Bag of Words, Document-Term Matrix or Document Feature Matrix. According to Manning and Schütz, the vector space model is one of the most common methods used in information retrieval because of its simplicity and the ability to represent documents as spatial relationships.[5] This is the motivation for using it here as well. However, in addition to the metaphorical relationship between topic and vector similarity, the vector space model provides the necessary input to mathematically analyze documents as points in space. This has opened the possibilities of measurement to include traditional measures of distance as well as the NLP specific algorithms.

---

[3] Christopher D. Manning and Hinrich Schütze, *Foundations of Statistical Natural Language Processing*, 1st ed. (Cambridge, Mass: The MIT Press, 1999), xxxi.

[4] Minmin Chen, "Efficient Vector Representation for Documents through Corruption," *ArXiv:1707.02377 [Cs]*, July 7, 2017, 1, http://arxiv.org/abs/1707.02377.

[5] Manning and Schütze, *Foundations of Statistical Natural Language Processing*, 539.

In addition to the general principles of mathematics, there are many similarity algorithms with which to analyze vector space models. In 2013, the International Journal of Computer Applications published a comprehensive survey of 14 more common text similarity approaches.[6] Literature research also revealed many algorithm comparisons studies. Two highly cited studies related to the work in Chapter II was a study titled, "Similarity Measures for Text Document Clustering,"[7] and a study titled, "Impact of Similarity Measures on Web-page Clustering."[8] This thesis tests five algorithms that are the most applicable to the research here. While more algorithms are tested in this thesis than these two articles, both articles come to similar conclusions to what is presented in Chapter II.

Chapter III builds on the previous research and focus on improving algorithm performance by refining the input vectors. To do this, a separate algorithm called Term Frequency – Inverse Document Frequency (TF-IDF) is used. Much research suggested that TF-IDF can be used to quantify the vectors in terms of word significance instead of word frequency alone.[9] TF-IDF can also be used to extract keyword summaries. Chapter III uses TF-IDF to improve the quality of the input data and perform keyword extraction in Chapter III.

Another motivation of this thesis is data visualization, which is the focus of Chapter IV. The output of the algorithms must be displayed in a way that facilitates decision-making. The literature review for related research centered on combining the concept of vector similarity with means of displaying document similarity. The results of this research frequently pointed to the science of social network analysis. This makes sense because

---

[6] Wael H. Gomaa and Aly A. Fahmy, "A Survey of Text Similarity Approaches," *International Journal of Computer Applications* 68, no. 13 (April 18, 2013): 13, https://doi.org/10.5120/11638-7118.

[7] Anna Huang, "Similarity Measures for Text Document Clustering," *Proceedings of the 6th New Zealand Computer Science Research Student Conference*, January 1, 2008.

[8] Alexander Strehl, Joydeep Ghosh, and Raymond Mooney, "Impact of Similarity Measures on Web-Page Clustering," in *AAAI 2000* (AAAI Workshop on AI for Web Search, Austin: AAAI Press, 2001), 7, https://www.aaai.org/Papers/Workshops/2000/WS-00-01/WS00-01-011.pdf.

[9] Hans Christian, Mikhael Agus, and Derwin Suhartono, "Single Document Automatic Text Summarization Using Term Frequency-Inverse Document Frequency (TF-IDF)," *ComTech: Computer, Mathematics and Engineering Applications* 7 (December 31, 2016): 286.

here, as in social network analysis, one needs to visualize important relationships between objects. According to Cunningham, Everton, and Murphy, Social Network Analysis focuses on how interaction patterns better explain actor behavior than their attributes.[10] This view of social interactions is the closest analogy to what is hoped to be accomplish here. Chapter IV is most concerned with depicting the position of reports relative to their position in the collection. As in Social Network Analysis, clusters of reports are related by topic in the same way clusters of people are related by clique; some reports are more central to the topic, as some actors are more central to the social network; and some documents connect multiple topics as some actors are mediators between multiple groups of people.

No literature has been found of a similar system of combining these concepts to address the problems of unstructured analytics, storage, and retrieval. This thesis aims to fill the gap in the existing research, by examining the feasibility of combining these models into a unified system as described here, and testing the utility of such a system to address the military problem of operational reporting.

E. THESIS ORGANIZATION

This thesis is constructed three sections; each corresponding to a component of the process. In Chapters II and III, we provide research toward answering each of the two research questions. Chapter II focuses on semantic similarity. We researched a system of quantifying the topics contained in news articles in order to measure the similarity between like articles. These results were used to devise a method of automatically grouping reports based on semantic similarity. This is more advanced than grouping reports on things like search terms or labels such as dates and place. The purpose is to research a method in which a document is searched based on the meaning of the content. Chapter III builds on the concepts of the previous chapter and researches a method of labeling the topics of documents based on the importance of the words they contain. This is a process called keyword extraction. This is an automated process that extracts important terms from a

[10] Daniel Cunningham, Sean Everton, and Philip Murphy, *Understanding Dark Networks: A Strategic Framework for the Use of Social Network Analysis*, Reprint edition (Lanham: Rowman & Littlefield Publishers, 2016), 8.

single document, but more importantly determines the most important terms in a document relative to a collection of documents. The labels provide semantic context to mission planners. Also, labeling the documents in this way will be important to build the metadata for storage and retrieval. Chapter IV will then build on the results of Chapters II and III and demonstrate a prototype of a system that uses these two concepts as an engine for analyzing and visualizing unstructured data.

THIS PAGE INTENTIONALLY LEFT BLANK

II. MEASURING SIMILARITY USING STATISTICAL METHODS

In this chapter we examined a method of automatically grouping documents by topics, a process called semantic clustering. Statistical-NLP methods were used to convert an archive of 2,225 pre-labeled news articles into measurable elements, called vectors. The similarity of the vectors was then measured using five popular similarity algorithms; in this chapter, a similarity algorithm called the Jaccard similarity performed the best. The results revealed a statistically significant correlation between each similarity measure and the article's news category—the articles were more similar to articles within their category than to the other articles. This similarity illustrates that the computer was able to accurately derive meaning from the articles. Similarity is an essential component of a system that can be used to organize unstructured reporting. We will build on these results in the following chapters. The details of this research are provided here to demonstrate the process and provide evidence of the results. See annex A for the complete code used to conduct this analysis.

A. HYPOTHESIS AND OVERVIEW

To research semantic clustering, we measured the topics contained in a collection of text documents. The documents consisted of 2,225 news articles that were published between 2004 and 2005 by the BBC News service.[11] The topics were randomly selected and all of the article filenames were pre-coded according to their major news category. The news categories and the respective codes are Business (b), Politics (p), Technology (th), Sports (s), and Entertainment (e).

The articles were measured using several techniques for measuring semantic similarity. The null hypothesis (H_0) is that no measurable semantic relationship among the article groupings exists. The alternative hypothesis (H_a) is the opposite; there is a

[11] D. Greene and P. Cunningham, "Practical Solutions to the Problem of Diagonal Dominance in Kernel Document Clustering," in *ICML2006* (23rd International Conference on Machine learning, New York: ACM Press, 2006), 377–84.

measurable semantic relationship between the article categories. The similarity metrics served as the independent variables, and the category matches (True/False) were the dependent variables. A logit regression model was used because of the dichotomous dependent variable. A total of 2,474,200 measurements per metric were taken to measure the similarity between each pair of articles. A statistically significant correlation between the two variables provides support for the alternative hypothesis. This means that the algorithms used were able to detect the true similarity of the article topics within their news category, at a rate better than chance. The R programming language was used for all processing steps.[12] The R libraries used were Quanteda,[13] Gtools,[14] Tidyverse,[15] and Textstem.[16] An R library called stargazer was used to calculate the statistics and create the regression table used in the results.[17]

B. DATA PROCESSING AND PREPARATION

The BBC News articles were first read into an R data frame. A data frame is a data structure similar to a spreadsheet; each row is called an observation, and each column is a variable. The data frame contained 2,225 observations, one for each article, and two variables: the document filename and the corresponding article's text. For example, observation 1 consists of filename = "b (1).txt" and text = "Ad sales boost for time warner…" The entire collection of texts is called the *corpus*.

Once data loading was complete, the corpus was pre-processed in order to standardize each article's contents. The basis for the analysis is the measurements of the common words in the articles. However, we are only interested in analyzing the most

---

12 R Core Team, *R*, version 3.6.3 (Vienna, Austria, 2020), https://www.R-project.org/.

13 Kenneth Banoit et al., *Quanteda: An R Package for the Quantitative Analysis of Textual Data.*, version 2.0.1, 2018, https://quanteda.io.

14 Gregory R. Warnes, Ben Bolker, and Thomas Lumley, *Gtools: Various R Programming Tools*, version 3.8.2, 2020, https://CRAN.R-project.org/package=gtools.

15 Hadley Wickham et al., *Tidyverse: Welcome to the Tidyverse*, version 1.3.0, 2019, https://CRAN.R-project.org/package=tidyverse.

16 Tyler W. Rinker, *{textstem}: Tools for Stemming and Lemmatizing Text*, version 0.1.4 (Buffalo, NY, 2018), http://github.com/trinker/textstem.

17 Marek Hlavac, *Stargazer: Well-Formatted Regression and Summary Statistics Tables.*, version 5.2.1, 2018, https://CRAN.R-project.org/package=stargazer.

important words. To obtain the best results, we first removed *stop words* such as "and," "a," and "the." These words were very common in the corpus but did not add any useful meaning beyond the language structure.[18] During this process, we also removed URLs such as "http://www.bbc.com," punctuation and numbers, and converted all words to lowercase. This was done automatically with the Quanteda library.

We next performed *stemming* and *lemmatization* on the corpus. Stemming is the process of removing word prefixes and suffixes in order to measure a common root word.[19] For example, words such as "attacked," "attacking," and "attacker" should be reduced to the root word "attack." Without stemming the algorithm would measure these as three distinct words.

Lemmatization is a more sophisticated method of simplifying words to their lemma, which is their base dictionary meaning.[20] For example, the forms of the word "run" and "ran" have the same lemma, "run." It is logical to understand that these three words are related, and verb tense is not important for our purposes. This is a good place to point out that custom military stemmers and lemmatizers will likely improve the performance of the algorithm when used with military reports. The standard tools may have problems with documents that contain military acronyms, military terminology, or phonetic spellings of foreign terms. In this experiment, we used the common dictionary in a popular open-source R library, textstem, to perform the stemming and lemmatization.[21]

18 Bhargav Srinivasa-Desikan, *Natural Language Processing and Computational Linguistics: A Practical Guide to Text Analysis with Python, Gensim, SpaCy, and Keras*, 1st ed. (Birmingham - Mumbi: Packt Publishing, 2018), 19.

19 Srinivasa-Desikan, 47.

20 Srinivasa-Desikan, 47.

21 Rinker, *{textstem}: Tools for Stemming and Lemmatizing Text*.

Table 1. DFM Showing the First 6 Rows and First 10 Columns

docs	ad	sale	boost	time	warner	profit	quarterli	us	medium	giant
b (1).txt	1	5	2	3	4	10	1	3	1	1
b (2).txt	0	0	0	0	0	1	0	2	0	0
b (3).txt	1	0	0	2	0	0	0	0	0	0
b (4).txt	0	0	0	1	0	0	0	0	0	0
b (5).txt	0	0	0	1	0	0	0	7	0	0
b (6).txt	0	0	0	0	0	2	0	2	0	0

The final step in this phase is called *tokenization*. The actual measurements are performed on a matrix consisting of the words in the corpus. Each word in a corpus is called a token and this matrix is called a Document Feature Matrix (DFM). In this experiment, the tokenization produced a DFM consisting of 2,225 rows that corresponded to each article and 22,221 columns that corresponded to all unique words (i.e., tokens) that occur in the corpus.

The DFM is a vector space model with each article represented as a vector of tokens using real numbers in the Euclidean n-space (R^n); this simply means we reshaped the articles in a way that can be quantified and measured. We base measures of semantic similarity on measures of vector similarity.[22] These vectors are numeric representations of the article's meaning. Each element (a_{ij}) in the DFM represents the number of times word j (columns) appears in article i (rows). For example, in Table 1, the value in row one, the article *b (1).txt*, and column two, the word "sale," equals five ($a_{1,2} = a_{b\ (1).txt,\ sale} = 5$). This merely means the word "sale" appears five times in that article. This count is referred to as the Term Frequency (TF). It is logical that other articles containing a high frequency of the word sale will be more similar to this article than those that do not. The entire collection (i.e. DFM row) of the TF measures form the article's vector. Again, the vectors in the DFM are just a method of representing the articles' meaning in a way that can be measured. **This concept is the basis for quantifying the semantics of a document.**

22 Manning and Schütze, *Foundations of Statistical Natural Language Processing*, 296.

We can also get a sense of the article's topic just from visually inspecting the contents of its vector. This is the point at which the machine is designed to mimic the human reader's perception. In Table 1, the words "sale," "time," "warner," and "profit" occurred most frequently in the article b (1).txt, an occurrence of 5, 3, 4, and 10, respectively. Intuitively, these high-frequency words give us a good idea of the article's news category of business as well as the article's topic. The lead sentence from article b (1).txt is, "Quarterly profits at U.S. media giant Time Warner jumped 76% to $1.13bn (£600m) for the three months to December, from $639m year-earlier."[23] We will revisit this concept using Term Frequency – Inverse Document Frequency (TF-IDF) in the next chapter. Also, note in Table 1, row 1, there are no stop words, and the lemmatizer changed some variations of words. The lemmatizer changed "media" to its singular form "medium." This allows our algorithms to focus on words that are at the core of the article's topic.

C. DATA ANALYSIS

In the data analysis phase of the research, we measured the similarity between all of the article vectors in the DFM. To perform the analysis, we used eight NLP models that consisted of five measurements of similarity. The measurements used were simple matching, Dice; Jaccard; and cosine similarity measures, as well as Euclidean distance.

The simple matching algorithm comes from the field of taxonomy, which is a field of science that deals with the classification of organisms.[24] The simple matching algorithm, also called the Sokal & Michener distance measure, is the ratio between the count of matching vectors in a document, both **zero and non-zero**, and the count of all of the vectors.[25] This means a match is counted once if both documents contain the same

[23] "Ad Sales Boost Time Warner Profit," BBC, February 4, 2005, http://news.bbc.co.uk/2/hi/business/4236959.stm.

[24] "Definition of Taxonomy," Merriam-Webster, 2020, https://www.merriam-webster.com/dictionary/taxonomy.

[25] Seung-Seok Choi, Sung-Hyuk Cha, and Charles Tappert, "A Survey of Binary Similarity and Distance Measures," *Journal of Systemics, Cybernetics and Informatics* 8, no. 1 (March 10, 2011): 43–44. See formula number (7) on page 44. In formula (7), a = (1,1), b = (0,1), c = (1,0), and d = (0,0); see page 43, for the explanation of the variables.

word, or similarly counted if they do not contain the same word; this sum is then divided by the count of all of the words in the corpus, see Equation (1.1).

$$\text{Simple matching} = \frac{\sum M_{0,0} + \sum M_{1,1}}{|V|} = \frac{|C \cup D|}{|V|} \qquad (1.1)$$

Equation 1.1 is expressed here two in two ways for clarity. In the first Equation, $M_{0,0}$ represents a zero-vector match and $M_{1,1}$ represents a non-zero vector match. In the second Equation set C and D represent the two disjoined sets of matching zero and non-zero vectors. Here the "||" symbols indicates that this is the count of elements in a set, also called the cardinality.[26] V is the set of all elements contained in the DFM, which is the count of all tokens in the corpus.

The similarity measure ranges from 0 to 1; a score of 0 indicates no matches, and 1 would be all token matches or a duplicate article. Before we move on, and to gain the best understanding of this process, we should consider the added benefit that stemming and lemmatization have on the matching algorithms. Word matching is particularly useful because we removed any inflections and reduced the tokens in the corpus to the dictionary form of the words. Also, while Simple Matching is a basic measure of similarity, it is far more effective than keyword searches and less time consuming than manually reading documents for a similarity comparison.

The Jaccard and Dice similarities are similar to Simple matching, but only **non-zero vectors** are counted as a match. The Jaccard coefficient is the ratio of the count of common tokens in two documents, and the entire set of tokens contained in both documents. The more common words that exist, the greater the proportion. It is defined as the intersection over union where A and B are non-zero vectors, see Equation (1.2).[27]

---

[26] "Definition of Cardinality," Merriam-Webster, 2020, https://www.merriam-webster.com/dictionary/cardinality.

[27] Ramon F. Brena, *Quantitative Semantics and Soft Computing Methods for the Web: Perspectives and Applications*, ed. Adolfo Guzman-Arenas, 1 edition (Hershey, PA: IGI Global, 2011), 31.

$$\text{Jaccard coefficient} = \frac{|A \cap B|}{|A \cup B|} \tag{1.2}$$

$$\text{Dice coefficient} = \frac{2|A \cap B|}{|A| + |B|} \tag{1.3}$$

The Dice coefficient is defined as the ratio of twice the count of common tokens and the sum of the count of elements in A and B. This is defined as the ratio of twice the intersection over the sum of the cardinalities of A and B, see Equation (1.3). The difference between the Jaccard and Dice similarities is the Dice similarity controls for the length of the documents.[28] Like Simple Matching, the Jaccard and Dice similarities measures range from 0 to 1, where 1 is a perfect match and 0 is no shared similarity.

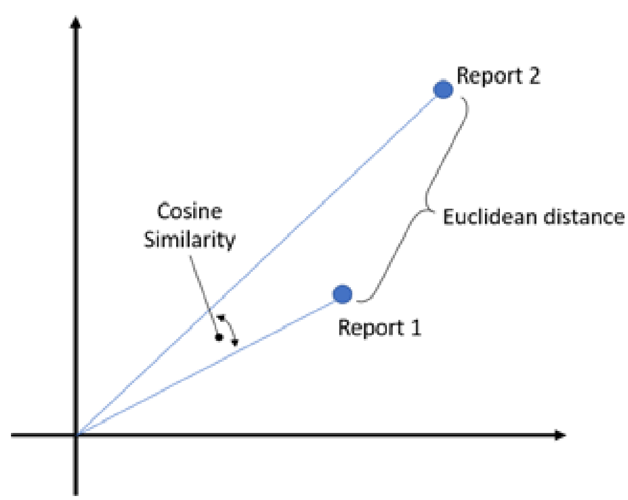

Figure 1. Cosine similarity vs Euclidean distance

The cosine similarity and Euclidean distance algorithms compare the similarity of two articles in the Euclidean vector space (see Figure 1). We can conceptually visualize the vector of each document as similar to points on a map. The cosine similarity measure is based on the cosine of the articles' vector on this plane. The similarity of article pair

[28] Manning and Schütze, *Foundations of Statistical Natural Language Processing*, 299.

increases as the angle between them decreases.[29] The cosine measure moves in the opposite direction of the magnitude of the angle; a cosine of two orthogonal, or right-angle vectors, would be cosine (90°) = 0, and two vectors with no angular difference would be, cosine (0°) = 1. Conceptually, this can be thought of as viewing two points from a central location. The angle between the two points with respect to the viewer is their cosine measurement (see cosine similarity, Figure 1). If these two points are in a direct line with each other, regardless of their respective distances from the observer, their degree difference is 0° and their cosine measurement is equal to 1. As in the other measures, cosine similarity measures range from 0 to 1, where 1 is a perfect match or duplicate article, and 0 is no shared similarity. It is important to note that the cosine similarity is normalized. This means the cosine similarity measurement is not affected by the document's length; restated in terms of our example, the observer's distance from the points will always be treated as the same. The cosine similarity of the two vectors is defined in Equation (1.4), where u and v are vectors in $\mathbf{R}^n$; the cosine similarity is the ratio of the product of the two vectors u and v, called the dot product of u and v, and the product of their lengths, or norms (i.e., $\|u\| \cdot \|v\|$).[30]

$$\cos\theta = \frac{u \cdot v}{\|u\| \cdot \|v\|} \tag{1.4}$$

Euclidean distance represents the distance between two points in the vector space. We can visualize this as simply the distance between two points on a map (see Euclidean distance, Figure 1). The closer two points are on land, the more similar terrain features they are likely to have. This is the same logic here. Just to restate for clarity, the cosine similarity algorithm measures the angle between two points and the Euclidean algorithm measures the distance. The Euclidean distance is defined in Equation (1.5), where a and b are two non-zero vectors.[31]

---

29 Brena, *Quantitative Semantics and Soft Computing Methods for the Web*, 31.

30 Brena, 31.

31 Steven Leon, *Linear Algebra with Applications*, 8th ed. (Upper Saddle River, NJ: Pearson Prentice Hall, 2010), 199.

$$\left|a - b\right| = \sqrt{\sum_{i=1}^{n} \left(a_i - b_i\right)^2} \tag{1.5}$$

It is important to note that Euclidean distance is not normalized. Unlike cosine similarity, the Euclidean distance measurements are affected by the length of the articles. Since the Euclidean algorithm measures distance, a zero indicates a perfect article match.

Table 2. Regression Table

		Dependent Variable: Article matches? (True / False)							
	Model (Logit):	1	2	3	4	5	6	7	8
Independent Varibles	Simple Matching	60.929***							161.929***
	(Standard Error)	0.499							1.041
	Euclidean Distance		-0.008***				-0.012***		0.016***
	(Standard Error)		0.000				0.000		0.000
	Cosine Similarity			3.871***			4.247***		1.107***
	(Standard Error)			0.017			0.018		0.026
	Jaccard Similarity				46.139***			295.431***	293.595***
	(Standard Error)				0.084			3.235	3.294
	Dice Similarity					27.345***		-147.885***	-146.776***
	(Standard Error)					0.050		1.913	1.948
	Constant	-61.557***	-1.062***	-3.467***	-5.236***	-5.581***	-3.184***	-3.349***	-164.552***
	(Standard Error)	0.493	0.004	0.010	0.008	0.008	0.010	0.025	1.041
	Pr(>|z|)	<2e-16***	<2e-16***	<2e-16***	<2e-16***	<2e-16***	<2e-16***	<2e-16***	<2e-16***
	RMSE	0.40044	0.40127	0.39734	0.36389	0.36407	0.39542	0.36349	0.36009
	Observations	2,474,200	2,474,200	2,474,200	2,474,200	2,474,200	2,474,200	2,474,200	2,474,200
	Log Likelihood	-1,238,880.0	-1,243,766.0	-1,221,424.0	-1,052,213.0	-1,053,502.0	-1,214,242.0	-1,049,125.0	-1,032,141.0
	Akaike Inf. Crit.	2,477,764.0	2,487,537.0	2,442,852.0	2,104,431.0	2,107,007.0	2,428,491.0	2,098,255.0	2,064,294.0

Note: *p<0.1; **p<0.05; ***p<0.01

D. RESULTS

We used logistic regression to evaluate the performance of competing NLP algorithms. A logistic regression is a standard method of evaluating the probability of a relationship between an independent variable and a binary dependent variable.[32] In this experiment, the relationship being measured is the correlation between the article similarity, calculated using each of the five similarity algorithms, and positive category matches. The Regression Table depicted in Table 2 lists the results of the models. Models one though five are single algorithm models. Model number six is composed of the cosine

32 "Logistic Regression," Encyclopedia of Mathematics, March 12, 2016, https://encyclopediaofmath.org/wiki/Logistic_regression.

similarity and Euclidean distance, model seven is composed of the Jaccard and Dice similarity, and model eight is composed of all five metrics. These results indicate that our hypothesis was correct; using the system presented here, there seemed to be a statistically significant correlation between each of the similarity measures and the article categories. Model eight had the lowest Akaike's Information Criteria (AIC) and Root Mean Squared Error (RMSE), and the highest log likelihood. This indicates that model eight was the best fit. The Jaccard similarity, model four, was the best fit for a single algorithm and even provided a better fit than model six – the Euclidean distance and cosine similarity combined. Also, the regression coefficients indicate the expected relationship between the independent and dependent variables for each of the models. In each of the similarity measures there is a positive relationship between the dependent and independent variables, which indicates that the algorithm was able to measure the contents of the articles. For example, the Jaccard similarity model has a positive regression coefficient, which indicates that the likelihood of matches increases as the Jaccard Similarity increases (see Figure 2). The only negative relationship is in model two (see Figure 3). This is expected because this model measures distance instead of similarity. As the Euclidean distance between two documents increases, the likelihood of a match will decrease. This behavior further demonstrated that this process worked as hypothesized.

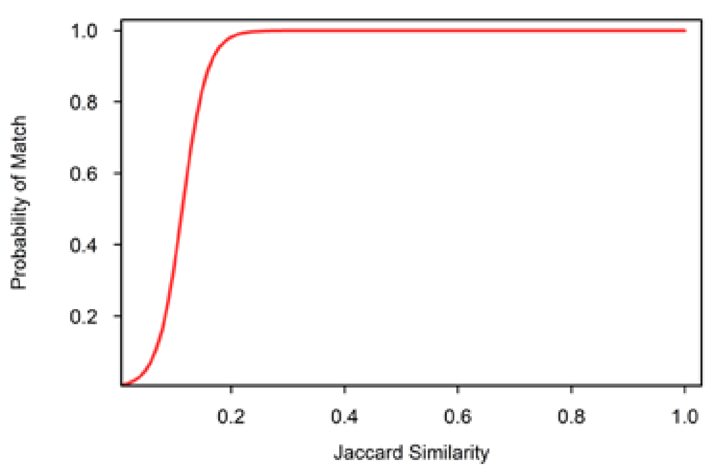

Figure 2. Jaccard Similarity Regression

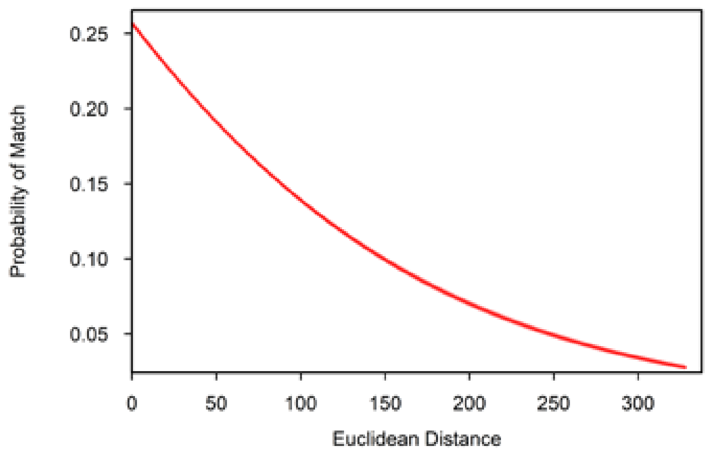

Figure 3. Euclidean Distance Regression

E. CONCLUSIONS

This chapter shows how NLP algorithms can be used to measure similarity in reports, which directly addresses research question number 1. These results demonstrate that our process was able to correctly measure the similarity of a collection of news articles. We tested the performance of multiple metrics and analyzed the results. Similarity is a step in a process of addressing the problems data analytics, data storage, and data retrieval. We can use these algorithms to create a system that acts as report-based search engine. Using similarity, the computer essentially can pre-screen reports and create clusters of topics. The clusters then can then be visually analyzed as document networks, which is much more efficient than manually reading though a collection of documents. The research presented in this chapter is the foundation for all of the work that lies ahead.

THIS PAGE INTENTIONALLY LEFT BLANK

III. TOPIC MODELS USING TERM FREQUENCY – INVERSE DOCUMENT FREQUENCY

This chapter examines the Term Frequency – Inverse Document Frequency (TF-IDF) algorithm. TF-IDF is an NLP algorithm that can be used to extract important keywords contained in a document in order to create a statistical element called a topic model. Topic models provide an intuitive summary of a document's subject. TF-IDF also improves the effectiveness of the similarity algorithms. While the similarity process described in Chapter II provides a method of clustering commonalities in reports, TF-IDF removes known similarities and highlights important differences. This improves measurement accuracy and will provide the scalability necessary to conduct analysis at varying levels of command. Here we describe the TF-IDF process in detail, demonstrate the process of topic modeling, and demonstrate how TF-IDF can be used to improve similarity accuracy.

A. OVERVIEW

TF-IDF is a process that can be used to automatically extract the most important terms from a single report. More importantly, TF-IDF can be used to "weight" the most important terms in a report relative to the entire collection of reports, or the corpus; in other words, what is unique in one report in contrast to others in a collection. We will use TF-IDF in a process called *topic modeling*, which is a statistical method of summarizing the abstract concepts of a document. This process builds on the similarity models in Chapter II. The combination of the similarity process and topic models will provide the NLP suite for the interface demonstrated in Chapter IV.

The TF-IDF algorithm is a component of the set of algorithms used by many search engines. While the exact combination of algorithms is often considered a trade secret, there is a significant amount of research that suggests TF-IDF is the most popular weighting scheme. In a 2015 study of research-paper recommender systems (e.g. Google Scholar, Science Direct, and Springer Link), researchers noted that TF-IDF was used by 70 percent

of the systems.[33] Other systems in the same study used TF alone, which is a variant of TF-IDF. The prevalence of this algorithm by the top commercial search engines speaks to its usefulness in document retrieval. This algorithm can be repurposed to improve the utilization of operational reporting. We used the Quanteda R library to calculate the TF-IDF values in this thesis.[34]

Mathematically, TF-IDF is typically defined as the Term Frequency (TF) multiplied by the Inverse Document Frequency (IDF). TF is simply a measurement occurrence of similar terms in a single document. TF can be expressed as a **raw count** or a ratio. The raw count is simply the number of occurrences of a term in the document. The ratio is expressed in Equation 2.1, where i is the raw count of a term in a document and j is the total of all the terms of a document. The ratio controls for document length by returning the term as a fraction of the document. This experiment tested both methods. We will use the raw count here because the ratio did not significantly improve performance and resulted in very small TF-IDF values.

$$TF_{i,j} = \frac{i}{j} \tag{2.1}$$

The use of TF to extract meaning is based on the hypothesis that a high frequency occurrence of specific words predicts the meaning of a topic. In 1954, Hans Luhn, a pioneer in Computer Science, said, "the more frequently a notion [term] and combination of notions occur, the more importance the author attaches to them as reflecting the essence of his overall idea."[35] Luhn's concept is fairly easy to conceptualize within a single document. For example, a report on Afghanistan will likely contain high TF values of terms such as "Afghanistan," and even a higher TF for the root "afghan." In this case, the term afghan is a reasonable label for this report. But what is the significance of the word "afghan" when

---

33 Joeran Beel et al., "Research-Paper Recommender Systems: A Literature Survey," *International Journal on Digital Libraries*, July 26, 2015, 319.

34 Banoit et al., *Quanteda: An R Package for the Quantitative Analysis of Textual Data.*

35 H. P. Luhn, "A Statistical Approach to Mechanized Encoding and Searching of Literary Information," *IBM Journal of Research and Development* 1, no. 4 (October 1957): 315, http://ieeexplore.ieee.org/document/5392697/.

we have an entire collection of reports about Afghanistan? The TF alone is not a good description of the overall importance of any single report, as these are terms that we would intuitively expect to see in most if not all of the documents. This presents a foreseeable problem in groups of military reports. The level of report diversity will affect the usefulness of topic models and similarity measures. How can one system be scalable enough to use at varying echelons of command? This is where IDF becomes important.

IDF is a measure of an individual term's relevance in the scope of the corpus. It is expressed in Equation (2.2), where the number of documents in the corpus (N) and the number of documents containing the term (n).[36]

$$IDF_{N,n} = \log_{10} \frac{N}{n} \tag{2.2}$$

The IDF calculation acts as a counterweight to the TF.[37] If a term is present in all of the reports in a given corpus, the ratio of $\frac{N}{n}$ will equal 1. Since the log $1 = 0$, the IDF = 0, and the product of the TF x IDF will similarly equal zero for any value of TF. This solves the problem of high TF values for words that appear across a majority of the documents of the entire corpus. For example, a 500-word report in which the word Afghanistan occurs 12 times has a $TF_{(Afghanistan)} = 12$. This is a frequency of over 2 percent of the document. However, if there are 100 reports in a collection and each contain this term at least once the IDF will equal 0 ($IDF_{(Afghanistan)} = \text{Log}(100/100) = 0$). This behavior is desirable because in this corpus, the word Afghanistan does not give the reader any new information. However, if the same reports are then sent to a higher-level command, and combined with reports from Iraq, the IDF values will change and the term Afghanistan ranks higher. This makes sense, because Afghanistan becomes a more relevant description of the reports. The word Afghanistan as a label is necessary to distinguish between reporting from the two countries. In this way the algorithm adapts to the unstructured data

36 Christian, Agus, and Suhartono, "Single Document Automatic Text Summarization Using Term Frequency-Inverse Document Frequency (TF-IDF)," 289.

37 Christian, Agus, and Suhartono, 289. We use log base 10, however there are several variations of this formula in literature.

and meets the informational needs of the reader. The TF-IDF algorithm gives the scalability to provide consistent results across different levels of command.

B. TOPIC MODELS USING TF-IDF

Topic models are intended to help bridge the gap between the mechanical process and human intuition. To demonstrate this, topic models were created from 500 Middle East news articles from the Nexis Uni (www.lexis.com) news database. Each article was processed into a DFM from a Microsoft Word (.docx) file. Only the article bodies were analyzed and the headlines were ignored.

This study used the same process for the creation of the DFM as was described in Chapter II. Once that was complete, the TF-IDF algorithm was used to further refine the token values. Topic models were created by selecting the top five tokens based on their TF-IDF values, which represent the five most significant terms in the document. Five of those topic models were randomly chosen to be presented here along with a discussion of the article's contents (see Table 3). The nature of this process is subjective. The goal is to demonstrate the level of usefulness to the reader. Note the words contained in the topic model are presented as measured by the algorithm, and were not altered to improve readability.

Table 3. Topic Models

Topic Models				
1	**2**	**3**	**4**	**5**
explosion	roadside	whitmer	wow	herat
tariq	helmand	hate	wfp	case
lbci	explosion	michigan	hunger	numb
blast	bomb	white	nobel	kabul
gasoline	kill	supremacist	food	test

(1) Topic Model 1: Explosion, tariq, lbci, blast, gasoline

Article one is from Naharnet Newsdesk, an online news site and the headline is "2 Killed, Over 20 Hurt in Tariq al-Jedideh Fuel Tank Blast."[38] This article is about an explosion in the Tariq al-Jedideh district of Beirut. The article is citing the facts as reported on LBCI TV. An important detail of the article is it was initially reported that the tank contained diesel fuel, but it actually contained gasoline. The topic model is a clear representation of the article's contents and even accurately ranks the word "gasoline" higher than "diesel," which did not rank in the top five terms.

(2) Topic Model 2: Roadside, helmand, explosion, bomb, kill

Article two is from the *Hindustan Times*, and its headline is "5 People Killed, 9 Injured in Roadside Bomb Explosion in Afghanistan's Helmand Province."[39] This article is only 100 words and the headline alone illustrate a straightforward correlation between the article contents and the topic model.

(3) Topic Model 3: Whitmer, hate, Michigan, white, supremacist

Article three is obviously not an article about the Middle East, but was reported in the Middle East. It is from an Iranian state-controlled news source called Press TV. The headline of the article is, "13 Charged in Plots against Michigan Governor Whitmer, Government."[40] While a U.S. article was not expected to be in the corpus, this headline is a good representation of the article's contents. The article also states that the plot against the governor was from a white supremacist hate group. This topic model is a good representation of the article.

[38] "2 Killed, Over 20 Hurt in Tariq al-Jedideh Fuel Tank Blast," Naharnet, October 9, 2020, http://www.naharnet.com/stories/en/275670.

[39] "5 People Killed, 9 Injured in Roadside Bomb Explosion in Afghanistan's Helmand Province," Hindustan Times, October 10, 2020, https://www.hindustantimes.com/world-news/5-people-killed-9-injured-in-roadside-bomb-explosion-in-afghanistan-s-helmand-province/story-uZatFIDP4iCljgxd1wtvjK.html.

[40] "13 Charged in Plots against Michigan Governor Whitmer, Government," PressTV, October 9, 2020, https://www.presstv.com/Detail/2020/10/09/635968/US-Michigan-governor-domestic-terrorism-white-supremacist-.

(4) Topic Model 4: Wow, wfp, hunger, nobel, food

Article four is from a news outlet called Yerepouni News, and the story headline is "Food is the Best Vaccine Against Chaos; UN Food Agency WFP Wins Peace Nobel."[41] The article is about the UN World Food Program winning the Nobel Peace Prize as stated in the headline. In the article, the WFP Executive Director David Beasley is quoted several times saying the words, "Wow! Wow! Wow!" The word "Wow" is the top-ranking TF-IDF term, and also, one that is unlikely to be in any other news article in the collection. In this case the topic model captured the informational aspects of speech, which may be useful for military jargon and acronyms. This topic model is a good representation of the article's true contents.

(5) Topic Model 5: Herat, case, numb, kabul, test

Article five may be the least intuitive of all of the topic models. The article is from an Afghanistan news channel called TOLO News, and the headline is "77 New Covid-19 Cases Reported in Afghanistan."[42] The headline is referring to new COVID-19 cases in Afghanistan in the last 24 hours. However, the article itself is a summary of cases across Afghanistan at different times. The author also presents the cases in terms of cases per tests given. For example, in Herat there were 120 positive cases out of 270 samples tested. The cities of Herat and Kabul have the highest numbers and occur most frequently in the article. This explains the high frequency terms in the topic model.

The article does not contain the word "numb." It is the stemmed or lemmatized version of the words number, numbers, and numbered. These terms occur frequently in the article. This illustrates one of the problems of the topic model: the reduced terms are not always intuitive. This problem is created by the stemmer or lemmatizer. A body of text containing both the variation of number and the dictionary word numb, meaning without feeling, would be reduced to the same word. As pointed out previously, the stemmer and

[41] "Food Is the Best Vaccine against Chaos"; UN Food Agency WFP Wins Peace Nobel," Yerepouni Daily News, October 9, 2020, https://www.yerepouni-news.com/2020/10/09/food-is-the-best-vaccine-against-chaos-un-food-agency-wfp-wins-peace-nobel/.

[42] "77 New Covid-19 Cases Reported in Afghanistan," TOLO News, October 9, 2020, https://tolonews.com/health-166920.

lemmatizer will need to be calibrated to mitigate these problems. Despite these problems, open-source libraries will likely be the best starting point for dealing with the type of issues we see in this article. The popular libraries will work a majority of the time for common dictionary words. However, military terminology libraries will need to be created and maintained to obtain the best overall results.

C. IMPROVING SIMILARITY USING TF-IDF

To further test the accuracy of the TF-IDF topic models, we re-examined the results of the similarity experiment in Chapter II. All parameters of the experiment were the same. In this experiment, the DFM was further refined using the TF-IDF algorithm. In Table 4, model 4 the cosine similarity performed better than the Jaccard algorithm. While only the cosine and Jaccard models are presented here, all five models were retested with TF-IDF.

Table 4. Regression Table – Improved Performance Using TF-IDF

	Dependent variable: Article Matches? (True/False)			
Model (Logit)	**1**	**2**	**3**	**4**
Independent Variable				
Jaccard Similarity	46.139***			
	0.084			
Jaccard w/TF-IDF		43.849***		
		0.082		
Cosine Similarity			3.871***	
			0.017	
Cosine w/TF-IDF				52.193***
				0.103
Constant	-5.236***	-4.752	-3.467***	-2.658***
	0.008	0.007	0.010	0.003
Pr(>|z|)	<2e-16***	<2e-16***	<2e-16***	<2e-16***
RMSE	0.36389	0.36572	0.39734	0.35852
Observations	2,474,200	2,474,200	2,474,200	2,474,200
Log Likelihood	-1,052,213.00	-1,061,622.00	-1,221,427.00	-1,035,185.00
Akaike Inf. Crit.	2,104,431.00	2,123,249.00	2,442,857.00	2,070,375.00
Note:				*p<0.1; **p<0.05; ***p<0.01

In Chapter II, the Jaccard algorithm out-performed all of the other algorithms. However, the Jaccard algorithm with TF-IDF performed slightly worse. Recall from Equation 1.2 that Jaccard calculates similarity based on sets; it is defined as a ratio of intersection over union. Since TF-IDF zeros out some vectors, the intersection value will

decrease while the union remains unchanged. These changes will not be uniform across the corpus, which will introduce errors. However, the cosine similarity score is based on the actual vector values instead of binary matches. Cosine similarity measures improves with the vector quality. In Table 4, model 3, the cosine similarity with TF-IDF had the lowest AIC, RMSE, and the highest log likelihood. This indicates that cosine similarity is the best fit for the data.

D. RESULTS

These results demonstrate two very important things. First, the improvement seen in the match probabilities indicates that the TF-IDF, as opposed to TF alone, produces a better vector representation of the article (see Figure 4). The left shift in the cosine measurement indicates better responsiveness of the algorithm. These results add a quantitative basis to topic models discussed in the previous section of this chapter. Remember, the vector is merely a numeric representation of the derived meaning of a document. Since TF-IDF produces a more precise vector for measurement, it is logical to conclude that the TF-IDF is the better choice for the topic models. The second point is about algorithm performance. To calculate the values for 2,225 articles, the Jaccard similarity took approximately 20 minutes, while the cosine similarity completed the task in less than 7 seconds. While these times will vary based on a number of variables, all tests performed during the course of this research indicated a clear time advantage on the side of the cosine similarity. TF-IDF improves algorithm performance without the loss of accuracy.

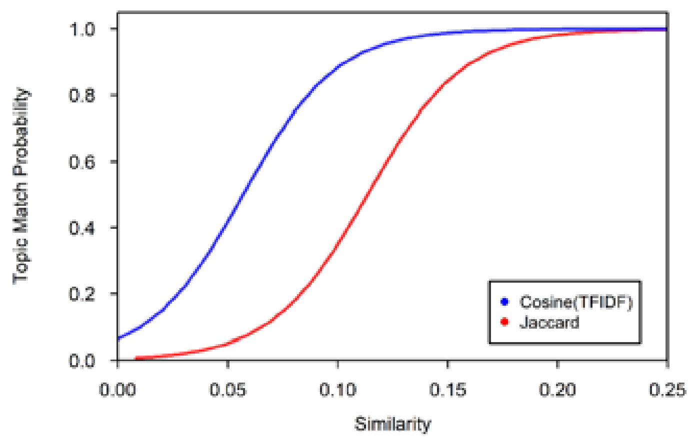

Figure 4. Comparison of Jaccard and Cosine with TF-IDF

E. CONCLUSIONS

This chapter demonstrates how NLP algorithms can be used to automatically summarize reports, which directly addresses research question number 2. In this chapter we used TF-IDF to build on the research of Chapter II. TF-IDF provides a basis for creating topic models, which can be used as text summaries. These metrics can also be used to increase the performance of the similarity algorithms. This improves the accuracy and speed of the measurement system. In the next chapter, we will make the connection between this theoretical work and a prototype of an analytical machine.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. DATA VISUALIZATION

This chapter's purpose is to demonstrate a method of visualizing the output produced by the algorithms in the previous chapters. The first research question was answered in Chapter 2, demonstrating that NLP algorithms can be used to organize unstructured reporting using similarity. Similarity provides an automated and scientific method of linking reports based on their meaning. Chapter III answered the second research question by demonstrating an automated method of creating report summaries called topic models. Topic models build on the work of Chapter II by adding context to the similarity measures, and scalability and accuracy to the algorithms. While the work up to this point has demonstrated possible solutions for improving analysis of operational reporting, this thesis would not be complete without demonstrating a prototype with which to visualize the results.

In this chapter we start by describing some foundational concepts of visualizing the data structures developed up to this point. To do this, we use a matrix of similarity measures from Chapter II and the topic models from Chapter III. We combine these two subjects to illustrate the framework for an interface. Then we use a corpus to demonstrate how the interface can be used to analyze a collection of news articles on topics relating to China. We apply this work to a series of "sub-topics" in that collection, which we will treat as three case studies. In this way, we address research question number 3: how can the results of NLP algorithms be displayed in ways that facilitate decision-making? I conclude this thesis by demonstrating a preliminary model of an analytical machine built from the theoretical work in the previous chapters. The goal is to advance the discussion towards a real-world working system.

A. PRELIMINARY CONCEPTS

Before moving on, it is beneficial to briefly describe the basis of the relationship between the data produced in the previous two chapters and, the visual representations used here. In this chapter we use graphs and graph theory to present the data. The concept is very intuitive; however, we need to first define some key terms and illustrate some core

concepts. A *graph* is a structure representing relationships between sets of objects.[43] Here the graphs are used as representations of a network of news articles, the objects, that are related by their topic similarity. Objects in a graph are called *vertices* or *nodes*. Relationships between the nodes are typically depicted as lines and commonly called, *edges*. Figure 5 depicts a graph with four blue nodes and three red edges. A graph structure is a natural representation of the matrix produced by the NLP algorithms in the previous chapters.

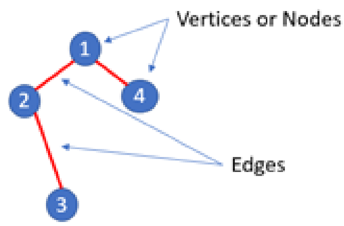

Figure 5. Illustration of a Graph

A matrix and its corresponding graph are illustrated in Figure 6. The matrix on the left, which is called a *weighted adjacency matrix*, contains the similarity measures for a collection of four reports. Adjacency means that the matrix represents connections between the nodes; when two nodes are connected by an edge, they are said to be adjacent. Weighted refers to the fact that the connections are represented by actual values instead of binary 1s or 0s, or True or False. The columns and rows, colored with a blue font, contain the names of the report, which become the nodes, and each red cell contains the similarity measure between the two corresponding reports, which becomes the edges. This matrix is the same format as the outputs of the work in Chapter II.

[43] Sriram Pemmaraju and Steven Skiena, *Computational Discrete Mathematics: Combinatorics and Graph Theory with Mathmatica.* (New York: Cambridge University Press, 2003), 10, https://www.cambridge.org/. Graphs in this sense are found in the field of discrete mathematics. See graph theory for more information on the science behind the study of the properties of graphs.

	Report 1	Report 2	Report 3	Report 4
Report 1	0	0.23	0.01	0.4
Report 2	0.23	0	0.25	0.2
Report 3	0.01	0.25	0	0.13
Report 4	0.4	0.2	0.13	0

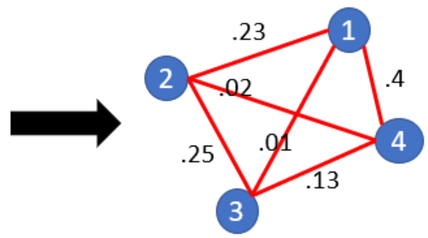

Figure 6. Matrix to Graph Translation

The grey diagonal cells, which are all zeros, are *self-loops*. This occurs when a report is measured to itself; self-loops will not be used here. This matrix is also said to be *symmetric*. Note, for example, that the similarity between reports 1 and 2 is 0.23 and it is actually measured twice; once between the two reports in the first row and the second column and again between the same two reports in the second row and the first column. All measurements occur twice in the matrix, but the values are only represented once in the graphs.

The graph representation of the matrix is on the right in Figure 6. Each blue node represents the four individual reports listed in the columns and rows, and the six red edges represent the corresponding values in the cells. Using graphs immediately gives a better sense of relationships between the reports.

In this chapter, the open source ORA-LITE software package developed by the Carnegie Mellon School of Computer Science was used to automatically generate the graphs from the matrix produced in Chapter II.[44] However, all of this work is platform independent. The weighted adjacency matrix produced by the code in the previous chapters is a standard format used in graphs—Appendix A, lists code by chapter. This matrix can be directly imported into graphing software packages or incorporated into a custom software package with a graphing interface. ORA was used here because it provides an interface for graph visualization that is broadly accessible.

---

44 Kathleen M. Carley, *ORA-LITE*, version 3.0.9.9.116, Windows 64-bit (Pittsburg, PA: Carnegie Mellon University, 2020), http://www.casos.cs.cmu.edu/projects/ora/.

B. CHINA NETWORK

The three graphs depicted in Figure 7 were produced using the methods described in the previous chapters. The images, labeled A, B, and C, are actually three versions of the same graph, which was generated from 100 news articles from the Nexis Uni (www.lexis.com) news database. Each news article was published in the year 2020 and contains the terms "China" and "Military." This graph uses the cosine similarity with the TF-IDF measure as described in the previous chapters. This collection will be referred to as the China network.

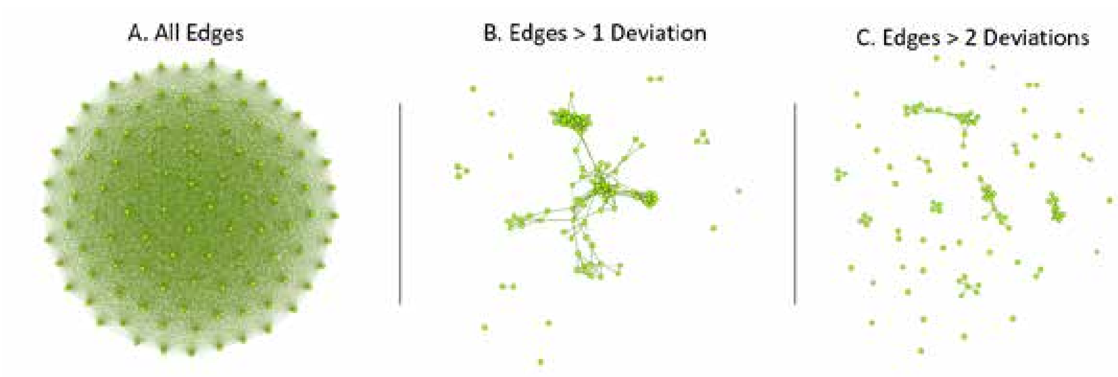

Figure 7. China Network

Recall from Chapter II that the similarity between every article is measured. Therefore, every node is connected to every other node on some measurable level. Initially, this produces an entirely connected graph (Figure 7A). For our purposes here, though, we only view edges that are statistically significant in terms of similarity. To do this we calculate the mean and standard deviation of the edge weights in the full graph. We then filter all edges less than two standard deviations from the mean. Values greater than two standard deviations from the mean are considered statistically significant. For the China network, this yields similarity values greater than 0.24.

In Figure 7 we can see that the network becomes progressively more fractured, from network B to C, as less similar edges are filtered out. There are also a number of articles without any ties (edges) to the other articles. These are called *isolates*. In this context,

isolates are articles that are not connected to any other article at a similarity level of 0.24. The isolates have been removed to simplify the network for our research here. In reality, these nodes have weaker connections and could be useful for an actual working analysis.

Filtering out nodes tied to other nodes at low levels of similarity is analogous to tuning the squelch on a radio or the declutter on a radar. While we filter at two standard deviations here, setting a lower similarity level would reveal weaker connections and could prove valuable in some analyses. There is no real definitive similarity level for filtering edges. Like the radar and radio, this analytical machine will produce noise. We must tune the system to discriminate the received information. This is analogous to using lower squelch on a radio in order to receive weaker signals. The weaker signals will be received along with background nose. The human operator must rely on his or her ability to separate the information.

Figure 8 depicts a close-up of the China network, only including edges greater than two standard deviations along with any resulting isolates removed. The edges have also been weighted by similarity value; thicker and redder edges indicate higher levels of similarity. Recall from Chapter II that a value of 0 indicates no similarity, and 1 indicates a perfect match. The illustration in Figure 8 represents a browsing view since it is not searching based on any one report. Reports are clustered into smaller groups based on their similarity. Each cluster of nodes represents a sub-topic within the collection of the Chinese article network.

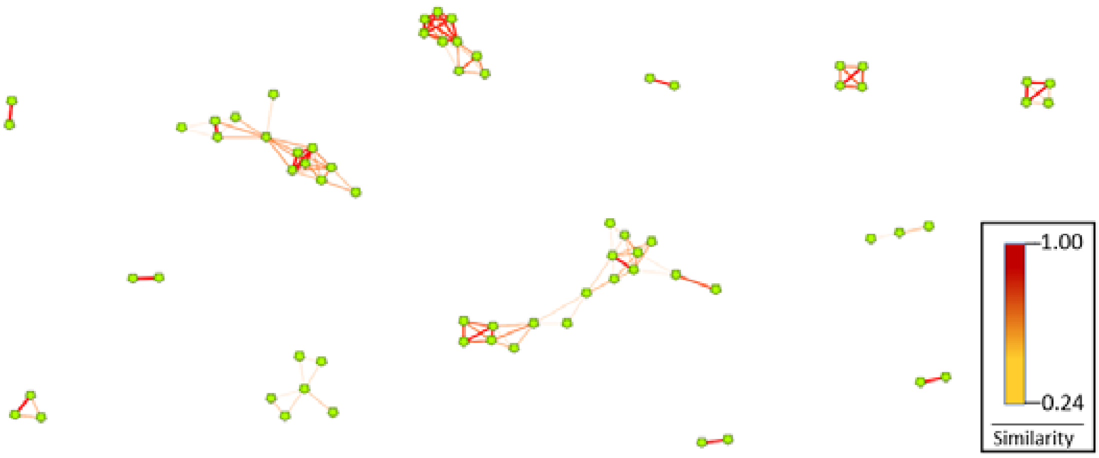

Figure 8. China Network with Isolates Removed

Before a more advanced analysis of the graph is discussed, it will help to first browse through a few details in the smaller clusters to get a feel for the network at the article level. Case studies 1 and 2 will focus on these smaller subnetworks, and case study 3 will examine mode advanced analysis.

1. Case Study 1: China Maritime Code of Conduct

Figure 9A presents a close up of the two-node graph depicted in the top left corner of Figure 8 with node and edge labels added. The high similarity score indicates that these documents are very similar articles. Figure 8 shows many nodes with thick red edges in the China network. This indicates that there may be at least five sets of duplicate documents and possibly many more contained within the larger clusters. Here the detection of the duplicates indicates that the algorithms performed as expected. Had this been an actual collection of reports, this information could also be used to trim the data. Alternately, if this system is put into operational use, a pair of duplicate documents containing randomly generated words can be used to calibrate and test the system.

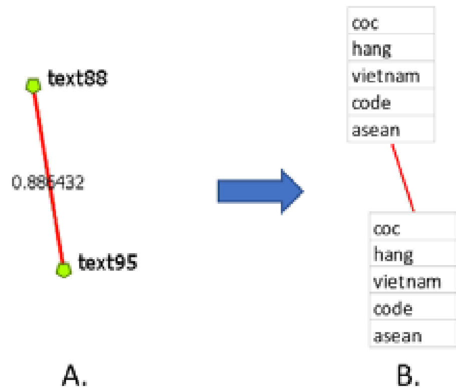

Figure 9. China Maritime Code of Conduct

Figure 9B adds the topic model summary to Figure 9A, and it shows that both of these articles are likely about Vietnam, the Association of South East Asian Nations (ASEAN), and codes, as in regulations; see Chapter III for topic models. Creating this entire network took less than a minute, and was entirely automated. More importantly we have uncovered a significant amount of information without opening a single document.

Next, the accuracy of the analysis is tested. A review of the corresponding documents reveals that the analysis was correct. Both articles in Figure 9 are essentially the same article; text 95 is a longer version of text 88. Both articles have the same headline, "Vietnam says China Military Drills Could Harm Maritime Code Talks." Text 95 was published by Channel News Asia.[45] Text 88 was published in the web edition of *the National Post*, a Canadian newspaper.[46] The articles described how China's military drills in the South China Sea complicated the maritime code of conduct (COC) talks, which were ongoing at that time. In the articles, the ASEAN foreign ministry spokeswoman Le Thi Thu Hang explained the situation to the reporters. Her name can be seen in the topic models. Recall from Chapter III that the algorithm ignores the headline; the similarity scores and topic models were derived entirely from the two articles' contents.

45 "Vietnam Says China Military Drills Could Harm Maritime Code Talks," Channel NewsAsia, October 1, 2020, https://www.channelnewsasia.com/news/asia/vietnam-beijing-south-china-sea-military-drills-13169338.

46 "Vietnam Says China Military Drills Could Harm Maritime Code Talks," *National Post*, October 1, 2020, https://nationalpost.com/pmn/news-pmn/vietnam-says-china-military-drills-could-harm-maritime-code-talks.

2. Case Study 2: South China Sea Militarization

The graph in Figure 10 is a close up of the four-node subnetwork in the top right corner of Figure 8. Even without the labels it is evident that texts 98, 23, and 33 are closer to each other than text 89. According to the topic models, these three articles are reporting about shipping, and, possibly maritime traffic. Once again, there seem to be duplicates; the similarity of texts 23, 33, and 98 is very close and has the same topic models; these will be visualized as one document. However, text 89 has a weaker connection to the three duplicate articles. The topic model indicates that the article in text 89 is discusses missile outposts. While text 89's connection is weaker to the remaining three articles, it must also be related to the topics contained in the other articles. It is logical to conclude that the missile outposts discussed in text 89 may have something to do with the shipping discussed in the other three articles (texts 23, 33, and 98).

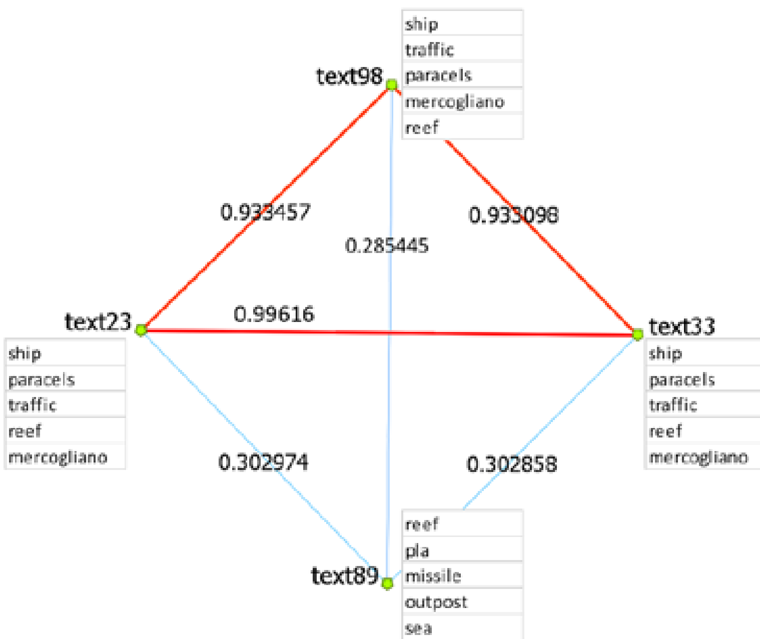

Figure 10. South China Sea Militarization

A review of the articles reveals that text 89 is from an article published on September 2, 2020, about a Pentagon report foreseeing a Chinese military buildup in the

South China Sea.[47] The article posits that China is building new outposts that will extend the operating area of PLA aviation forces. This is in addition to the installation of new guided anti-ship and anti-aircraft missile systems. The remaining articles, texts 23, 33, and 98, are dated a month later and provide additional information on this topic. These articles are from Asian news sources and discusses how maritime transmitter data reveals that commercial shipping is avoiding the South China Sea due to China's increased military presence.[48] Sal Mercogliano, a maritime historian at Campbell University in North Carolina states, "There's only a series of very narrow passages through the South China Sea straddling the Spratlys. It doesn't take much to cause a disruption in the supply chain, over those sea lanes."[49] This represents the kind of amplifying analytical relationship that can be visualized with this system. The additional information contained in text 23, 33, and 98 adds more information to the Pentagon report article (text 89). Despite the duplicates, these examples illustrate how the similarity measures, topic models, and the clusters can be combined in a way that improves analysis. In the next section, a larger subnetwork will be used to demonstrate more complex possibilities of investigation.

3. Case Study 3: China-India Border Dispute

Figure 11 depicts a much larger subnetwork than in the previous two examples. This subnetwork can also be seen at the center of Figure 8.

[47] "China's Military Set to Increase Presence in South China Sea: Pentagon," BenarNews, September 2, 2020, https://www.benarnews.org/english/news/philippine/Ch-SCS-09022020182723.html.

[48] "Data Shows Commercial Shipping Avoids Hotspots in South China Sea," Radio Free Asia, September 28, 2020, https://www.rfa.org/english/news/china/southchinasea-shipping-09282020155242.html.

[49] Codingest, "China's Build-up in the South China Sea Is Impacting Commercial Shipping," Mimicnews, September 28, 2020, https://mimicnews.com/chinas-build-up-in-the-south-china-sea-is-impacting-commercial-shipping.

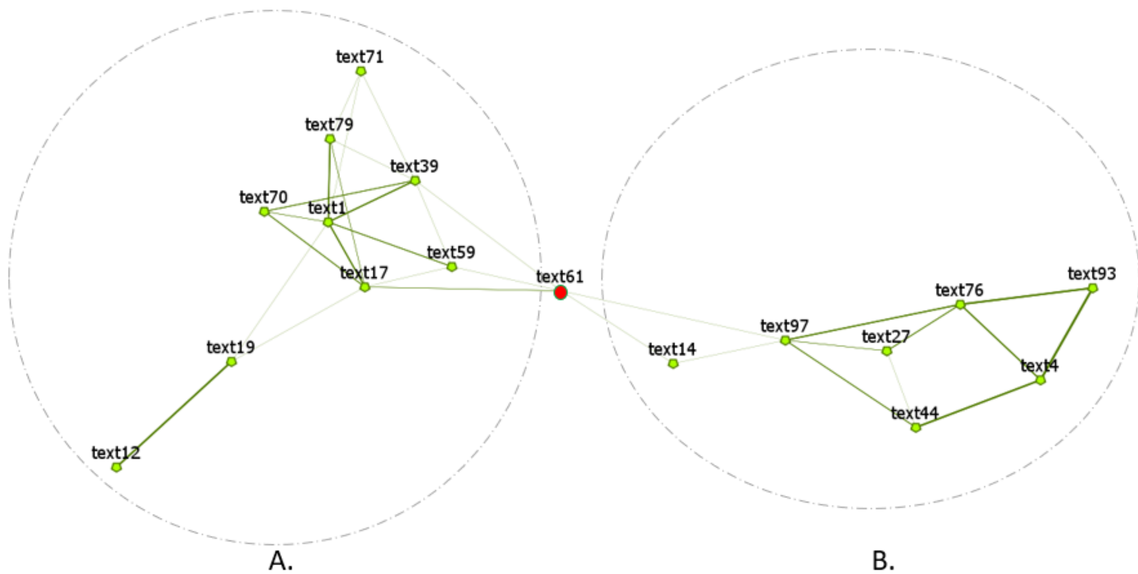

Figure 11. China-India Border Dispute

As in the previous cases, the graph in Figure 11 is only showing edges greater than two standard deviations from the mean. Additionally, all edges greater than 0.7 have been filtered out to minimize the duplicates that have been seen in the previous examples; recall that a similarity score of 1.0 is an exact copy. The edge thickness has been weighted by similarity; thicker edges represent closer nodes.

There are some important differences between this graph and previous examples. While the nodes are all connected, many do not have direct connections; this will be clarified below. This is an important concept that has not been discussed thus far, and would be extremely difficult to visualize without the graph. For example, there is no direct connection between text 12 in the lower left corner, and text 93 in the far right. However, there is a path between the two nodes. Therefore, on some level a semantic connection exists between these two articles. As seen in the topic models of Chapter III, this is the point in which the underlying mathematics connect with human intuition. There are many other nodes like this. In fact, there are so many such nodes, the network appears to be a connection of two distinct clusters. These subnetworks are labeled A and B and highlighted by the dashed circles in Figure 11.

Visually, it can be seen that any path from the left cluster A, to the right cluster B, must pass through texts 61 and 97. In graph theory, nodes such as these are said to have a high *betweenness centrality*. The pioneer of this concept, Linton Freeman, defined betweenness as the extent to which a node lies on the shortest path between pairs of other nodes.[50] According to Cunningham, Everton, and Murphy, had this been a social network, these individuals are said to be boundary spanners and potentially brokers of information.[51]

Betweenness values are calculated using Equation 3.1. This equation uses Freeman's original formulation, where $C_B(p_k)$ is the sum, or overall betweenness, of a point p_k; g_{ij} is the number of geodesics (i.e., shortest paths) linking points i and j; $g_{ij}(p_k)$ is the number of geodesics linking points i and j that contain p_k; n = the number of nodes.[52] ORA was used to calculate the betweenness scores for all nodes in the China network.

$$C_B(p_k) = \sum_i \sum_j \frac{g_{ij} p_k}{g_{ij}}; i \neq j \neq k \quad (3.1)$$

Text 61 has the highest betweenness centrality of the China network, which is represented by the red node in Figure 11. Think of this article as the search document. We want to find all topics related to this article; it is the starting point to explore both sides of this subnetwork.

Table 5. China Border Dispute Topic Models

	Text 12	Text 1	Text 59	Text 61	Text 14	Text 97	Text93
Topic Models	heliport	india	highway	border	kondapalli	singh	rafale
	airbases	eastern	india	shot	jaishankar	ante	ambala
	stratfor	ladakh	road	remote	temporary	lieutenant	rank
	border	intention	ladakh	indian	moscow	friction	officer
	defence	highway	helipads	jaishankar	truce	border	iaf

A. B.

50 Linton C. Freeman, "A Set of Centrality Based on Betweenness," *Sociometry* 40, no. 1 (1977): 34.

51 Cunningham, Everton, and Murphy, *Understanding Dark Networks*, 145.

52 Freeman, "A Set of Centrality Based on Betweenness," 37.

Table 5 lists the topic models for the boundary node, text 61, and the articles progressively father away from this point along a path to each extreme end of the subnetworks. Notice the symmetry between the topic models of the central node and the two endpoints (text 12 and 93). We can see that the word “border” ranks number 1 in text 61’s topic model. This term appears towards both ends of the subnetwork in text 12 and 97, but ranks lower. This indicates that the “border” topic is slowly becoming an artifact of these articles’ main topic as one moves away from the start point. The terms “heliport” and “Rafale” represent the primary topics of text 12 and 93, respectively. Although they are not pictured in Table 5, the topic models of 12 and 92 are also major topics of their respective closest neighbors, as is indicated by their thick edges in Figure 11.

From the topic models, it appears that the texts are centered around the ongoing border dispute between China and India. Since text 61 is a boundary spanner, the subnetwork is centered around this topic. The clusters, A and B, represent two distinct subtopics within this broader issue. Article 61’s headline is “India v China: Military Chiefs Meet in Bid to Defuse Boiling Cross-Border Tensions.”[53] This meeting was the sixth in a series of meetings between commanders on both sides. During the meeting, the Chinese complained about India building infrastructure on their side of the border. Infrastructure terms such as “highway” and “road” appear in the topic models of articles in subnetwork A.

The topic models of text 59 and 1, in Table 5, indicate that these articles are discussing infrastructure. In text 59, the author writes, “India has Doubled Down on its Work to Build a Bridge in the Himalayas as Tensions Reach a Boiling Point.”[54] Next, text 1 discusses the military buildup on both sides. The author states that India is developing

[53] Simon Osborne, “India v China: Military Chiefs Meet in Bid to Defuse Boiling Cross-Border Tensions,” *Daily Express*, September 21, 2020, https://www.express.co.uk/news/world/1338263/india-china-world-war-3-border-dispute-nuclear-armed.

[54] “India v China: New Delhi Risks Explosive Backlash in New Move That Will Infuriate Beijing,” *Daily Express*, September 29, 2020, https://www.express.co.uk/news/world/1341387/india-china-war-world-war-3-news-Ladakh-border-china-india-attack.

strategic border roads, bridges, and airfields.[55] The premise to the article is that each side is forcing the other's hand. While China complains about India's infrastructure, China's military buildup has left India with no other options but to strengthen its defenses.[56] The end article, text 12, is a *Hindustan Times* article outlining a report from Stratfor, a security intelligence consultancy.[57] The report states that China has essentially doubled its military positions since the 2017 standoff with India at Doklam. Since that time, China has built 13 new military bases, including airbases equipped with air defense systems.[58] Construction began on four of these installations after the latest border dispute with India at Ladakh.[59] The chain of progression of these topics can be seen, with each article amplifying an aspect of the subject in the previous article.

The path through the subnetwork B, from article 61 to text 93, follows an even more surprising chain of topics. In Figure 11 notice that two sequential articles, text 14 and 93, are not fully meshed inside of subnetwork B. Their topics are slightly closer to the topic contained in 61: The China – India border talks.[60] Text 14 is specifically discussing the details of the sixth round of border talks between the senior military commanders.[61] This round of talks focused on the tense military standoff in eastern Ladakh. The result of the talks was an agreement to discuss de-escalation sector by sector, but was not decisive.[62] The next node, text 97, is an article announcing the decision by military commanders to

---

[55] Subhash Kapila, "China and India in an Unprecedented 'State of War' in September 2020," *Northlines* (blog), September 28, 2020, http://www.thenorthlines.com/china-and-india-in-an-unprecedented-state-of-war-in-september-2020/.

[56] Kapila.

[57] Rezaul H. Laskar, "China Doubled Its Air Bases, Air Defences and Heliports near LAC in Three Years: Report," *Hindustan Times*, September 22, 2020, https://www.hindustantimes.com/india-news/china-doubled-its-air-bases-air-defences-and-heliports-near-lac-in-three-years-report/story-LgpbHDWgdfuh4cHwgyeQXP.html.

[58] Laskar.

[59] Laskar.

[60] Osborne, "India v China: Military Chiefs Meet in Bid to Defuse Boiling Cross-Border Tensions."

[61] Elizabeth Roche, "India-China Military Talks on LAC Disengagement 'Positive' but 'Inconclusive,'" MINT, September 22, 2020, https://www.livemint.com/news/india/india-china-military-talks-on-lac-disengagement-positive-but-inconclusive-11600782120799.html.

[62] Roche.

hold a seventh round of military talks.[63] This article is dated a week after the previous node, text 14. The article states that both sides have agreed to stop sending troops to the border and will proceed with a seventh meeting.[64] The last node in this chain, text 93, was written the day of the sixth round of the military talks discussed in text 14.[65] The article is discussing the Indian Air Force's (IAF) military show of force in Ladakh. According to the article, the IAF had readied its fleet of Dassault Rafale fighter jets, and was flying patrols in support of ground units that occupied key positions along the border. This was obviously done as a show of force during the commander's meeting, which is the central topic of discussion in this entire subnetwork.

In each subnetwork, it can be seen how the topics emanated from the border issue discussed in text 61, and progressively transitioned the discussion toward peripheral topics. These topics in themselves are central to each of their respective subnetworks, networks A and B. Notice that the edges contained within each subnetwork indicate a greater internal similarity. This means that each of the articles inside A and B will similarly provide further amplifying information for their respective topics: text 12 for subnetwork A, and 93 in subnetwork B.

This method of analysis can be applied to many datasets. For example, envision using an OPSUM, IIR, or, the Commander's guidance as the search document. Once the similarity is calculated, this document is selected as a starting point. The reader/analyst can then browse from this central location to discover more and more details related to the focus topic. This is a more accurate, and efficient, process than randomly reading though a collection of documents in the hopes of finding something useful.

Case study 3 begins to reveal the level of analytical efficiency that can be gained by visualizing similarity in this way. The China network contains 100 articles. The average article is 2 or 3 pages in length, which means roughly 200 to 300 pages of text. The cluster

---

63 "Seventh Round of India-China Military Talks on Border Row Likely next Week," *Hindustan Times*, October 1, 2020, https://www.hindustantimes.com/india-news/seventh-round-of-india-china-military-talks-on-border-row-likely-next-week/story-1DEyccDJrDufi9B6YTqKXM.html.

64 "Seventh Round of India-China Military Talks on Border Row Likely next Week."

65 "IAF Rafales Flying in Ladakh; India-China Military Talks Today," *Hindustan Times*, 21sep 2020.

of data in Figure 11 represents 17 articles, or roughly 35 to 50 pages. However, a good sense of this entire cluster can be obtained from the topic models and shape of the graph alone. This entailed reading a total of 35 words, which is less than a paragraph. Case study 3 is just the surface of the level of analysis that can be performed.

C. CONCLUSIONS

This chapter answers research question number 3, and more importantly melds the theoretical work in the previous chapters into a preliminary model of an analytical machine. It first demonstrated a process that can be used to visualize the outputs created by the similarity measures and the TF-IDF topic models. It then used the case studies as both a proof of concept and an introduction to using a graph to examine unstructured data. Visualization not only answers the research question, but it is the hope of this thesis that this prototype sparks the imagination, and serves as the inspiration for more advanced systems.

THIS PAGE INTENTIONALLY LEFT BLANK

V. CONCLUSIONS AND RECOMMENDATIONS

This thesis aimed to show that natural language processing can be used to improve the usefulness of military operational reporting. These results demonstrate a system of data mining and analysis from its theoretical foundations to a demonstration of its capability. This work addresses gaps in the current literature in that it integrates the work of document clustering, automatic text summarization, and practical visualization to address the military problem of unstructured data organization. From these results, one can see that clustering reports based on semantic similarity offers substantial advantages over current analytical procedures. Reports can automatically be processed and visualized in a way that reduces workload and captures meaningful relationships that would otherwise be extremely difficult to see; thus, improving military staff efficiency and analytics. However, these results describe more than a prototype. This is the foundation of a strategy for dealing with the rising complexities of unstructured data, which is a military-wide problem.

This work addresses the problem of unstructured data analysis on two levels. First it aimed to answer technical questions of unstructured information retrieval. These are represented by the theories, equations, and quantitative analysis in the previous chapters. Second, it hopes to take this concept further by illustrating a better methodology for unstructured information analysis.

In addition to an illustration of a prototype, this work aims to provide an approach to unstructured data analysis that could be broadly accessible across the joint force. Future work could build on these results by implementing the latest cutting-edge measurement and summarization algorithms into this same methodology. Regardless of the technical inner-workings, the concept of document clustering and graph-based report analysis could be implemented into a new data analysis skillset across the force. This is the real prize—a fighting force capable of managing the volume and speed of information in modern warfare.

A. FUTURE WORK

The findings presented here can serve as the basis for additional research in NLP, graph theory, machine learning, and unstructured data management. A major limitation of this thesis was the use of news articles instead of actual military reports. The news articles presented in this work were intended to reflect the type of topics that would be expected to be seen in a command, but could not replicate the uniqueness of military communication. This raises questions that require future research. At least three problems, relating specifically to military reporting, were discovered during the course of this research. These problems occur at the data processing and preparation phase, Chapter 2, Section B.

The first issue occurs during initial data cleaning. During this phase, stop words, numbers, and punctuation are removed. This can completely change the meaning of terms that occur as character-number combinations. For example, this will change words such as "MI-38," a Russian transport helicopter, to "MI," which can have many meanings such as the abbreviated state of Michigan or the NATO code for Malawi. This particular type of pattern occurs frequently in military documents.

Second, this system will need to read Military Grid Reference System (MGRS) coordinates. This will create both problems and opportunities. MGRS grids will fall in the category of the first problem described in this section—a character-number combination. There are many options for dealing with this issue. The algorithm does not alter the reports, and can be programmed to ignore all coordinates. Alternately, MGRS grids can be used by a sub-system that automatically connects reports to physical locations.

Third, it unclear how the standard lemmitizer and stemmer will work with military terms. This will be an especially difficult for transliterated foreign words such as Hezbollah, which can be written as Hizbullah or Hizballah. These words will likely be tokenized as separate terms, which will degrade the accuracy of similarity metrics and topic models.

Finally, this research raises many questions about using open source software on Secret Internet Protocol Router Network (SIPRNET) or Joint Worldwide Intelligence Communications System (JWICS) based systems. As stated at the beginning of this thesis,

one of the reasons that statistical algorithms were chosen is that they are open source and the internal workings of the algorithm is not proprietary. It was a goal of thesis to produce a prototype that a user can immediately field test on real world data. However, these systems will likely be classified and software installation restricted. The current DOD's open source software policy may be written in a way that stifles software research and development. Security is understandably a high priority. But the changing nature of the information environment demands equivalent shifts in information security policies. Research should be conducted on how to provide access to innovators, while mitigating information security risks.

B. FINAL THOUGHTS

This work is about more than the processing of documents. Whether the reports are SITREPs of military commanders in a regional command, or OPSUMs from Special Forces Operators on the forward edge of the battle area, the information from these military professionals is simply too valuable to waste. Human beings are the number one asset, and these documents represent communications with the Operator in the field. This is the most valuable information that we possess, and the old methods are limiting the effectiveness of our organization. The military is simply too large, manpower too small, and operations too numerous to continue to rely on archaic methods of analysis. Emerging technologies such as NLP offer important solutions to the growing intensity and complexity of modern warfare.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX

A. CHAPTER II CODE

```
################################################################################
# Title     : Chapter 2 - NLP Similarity Experiment
# Created by: George W. Bailey
# Created on: 4/9/2020
# Some portions of code derived from public sources.
################################################################################

################################################################################
# This code will reproduce the experiment in Chapter II.
# The code is broken down into sections by "----" and major sections by "####"
################################################################################


# See reference list for library citations.
library(quanteda)#analysis
library(gtools) #mixed sort
library(tidyverse)#map_df()
library(textstem) #stemming and lemmatisation

################################################################################
#-------------------------------Read Article Files------------------------------
# Note: uses standard text files as input.

setwd("c:/ADD_YOUR_DIRECTORY/") # You must add your directory!!!!!!!!

# read all articles
article_text <- list.files(pattern = ".txt")
article_text <- mixedsort(article_text) #sort the files sequentially

# create a dataframe with all of the files and add a new column with file names
# MagicMerge must be unloaded
article_text.df <- map_df(article_text, ~ data_frame(text = read_file(.x)),
                  mutate(fileName = basename(.x)))


#-----------------------------------------------------------------------------
# create a corpus from the unstructured text
article_corpus <- corpus(article_text.df)

# stem and lemmatize corpus
article_corpus <- stem_strings(article_corpus)
article_corpus <- lemmatize_strings(article_corpus)

# create tokens from corpus and remove any URLs
article_tokens <- quanteda::tokens(article_corpus, remove_url = TRUE)

# convert tokens to document frame matrix & remove stopwords
article.dfm <- dfm(article_tokens, remove = stopwords('en'))
```

```
#----------------------------------------------------------------
# The section of code is used in Chapter III when this experiment is repeated
# with TF-IDF metrics instead of term frequency.

# convert term frequency measures to to TF_IDF measures
article.dfm <- dfm_tfidf(article.dfm)

# Alternate version of above line.  Added here only for reference.
# Uses proportion of TF instead of count as stated in Equation 2.1.
# article.dfm <- dfm_tfidf(article.dfm, scheme_tf = "prop",
#                          scheme_df = "inverse", base = exp(1))
#----------------------------------------------------------------


################################################################################
#------------------Calculate Similarity Metrics-------------------------------

# Function to streamline creation of distance and similiarity metrics.
text_stat <- function(dfmat, ids, method='euclidean') {

    if (method %in% c('euclidean', 'manhattan', 'minkowski', 'canberra')) {
    mat <- as.matrix(as.dist(textstat_dist(dfmat, method = method)))
    val_name <- paste0('dist_', method)
  } else {
    mat <- as.matrix(as.dist(textstat_simil(dfmat, method = method)))
    val_name <- paste0('simil_', method)
  }
  df <- data.frame(
    article1 = ids[col(mat)],
    article2 = ids[row(mat)],
    val = c(t(mat)),
    stringsAsFactors = FALSE)
  df <- filter(df, article1 < article2)
  names(df)[3] <- val_name
  return (df)
}

# Uses text_stat function to create edge lists with distance metrics.
# Notice one is created for each metric.
edges.euclidean <- text_stat(article.dfm, ids=article_text, method = 'euclidean')
edges.cosine <- text_stat(article.dfm, ids=article_text, method = 'cosine')
edges.matching <- text_stat(article.dfm, ids=article_text,
                            method = 'simple matching')
edges.dice <- text_stat(article.dfm, ids=article_text, method = 'dice')
edges.jaccard <- text_stat(article.dfm, ids=article_text, method = 'jaccard')

################################################################################
# The section of code from here below is only used to calculate regression
# table, and graphs in Chapter II.  Not necessary to calculate similarity
# measures.
################################################################################
library(MagicMerge) # MagicMerge library may not be publicly available.

# Combine edgelists into a single data frame
```

```
edges <- d_merge(edges.euclidean, edges.cosine, by=c('article1', 'article2'))
edges <- d_merge(edges, edges.dice, by=c('article1', 'article2'))
edges <- d_merge(edges, edges.matching, by=c('article1', 'article2'))
edges <- d_merge(edges, edges.jaccard, by=c('article1', 'article2'))

#fix column name in simple matching
colnames(edges)[colnames(edges) == "simil_simple matching"] <- "simil_matching"

# Determine matches in articles.
a1 <- edges$article1
a2 <- edges$article2

edges$match <-  0
edges$match[grepl("b", a1)  & grepl("b", a2)]  <- 1
edges$match[grepl("p", a1)  & grepl("p", a2)]  <- 2
edges$match[grepl("th", a1) & grepl("th", a2)] <- 3
edges$match[grepl("s", a1)  & grepl("s", a2)]  <- 4
edges$match[grepl("e", a1)  & grepl("e", a2)]  <- 5

edges$match_bin <- ifelse(edges$match != 0, 1, 0) #make a binary 0 or 1

#------------------------------------------------------------------------------
# save and load RDA file for later use; for reference and commented out here.
#save(edges, file="edges.rda")
#load("edges.rda")

# Important!!!-------------------------------------------------------------------
# Stargazer library with one of the previous libraries.
# Load only after this point in the code.
library(stargazer)

# Creates generalized linear models - logistic model.
# Once for each model.
m1 <- glm(match_bin ~ simil_matching, data = edges, family = 'binomial')
m2 <- glm(match_bin ~ dist_euclidean, data = edges, family = 'binomial')
m3 <- glm(match_bin ~ simil_cosine, data = edges, family = 'binomial')
m4 <- glm(match_bin ~ simil_jaccard, data = edges, family = 'binomial')
m5 <- glm(match_bin ~ simil_dice, data = edges, family = 'binomial')

# Note these are combined models.
m6 <- glm(match_bin ~ dist_euclidean + simil_cosine, data = edges,
        family = 'binomial')
m7 <- glm(match_bin ~ simil_dice + simil_jaccard, data = edges,
        family = 'binomial')
m8 <- glm(match_bin ~ dist_euclidean + simil_cosine
        + simil_matching + simil_dice + simil_jaccard, data = edges,
        family = 'binomial')

#Create Regression Table-------------------------------------------------------

# Root mean squared error
rmse <- function(x, digits=5) {
  pred <- predict(x, type='response')
  res <- pred - x$y
  e <- sqrt(mean(res^2))
```

```
  re <- round(e, digits=digits)
  return(re)
}

# Fixing variable order and adding extra lines
stargazer(m1, m2, m3, m4, m5, m6, m7, m8, type = 'text', out = 'Table 1.html',
          order = c('pop', '^lgdppc$', '^polity2$','\\(pol','urban'),
          add.lines = list(c('RMSE', rmse(m1), rmse(m2), rmse(m3),
                             rmse(m4), rmse(m5),rmse(m6),rmse(m7),rmse(m8))))

# Plots ##################################################################
#single model plot
x_plot(m1, xvar = 'simil_matching', ylab = 'Likelihood of Match',
       xlab = "Simple Matching",file = "matching plot.png")
x_plot(m2, xvar = 'dist_euclidean', ylab = 'Likelihood of Match',
       xlab = "Euclidean Distance",file = "euclidean distance plot.png")
x_plot(m3, xvar = 'simil_cosine', ylab = 'Likelihood of Match',
       xlab = "Cosine Similarity",file = "cosine plot.png")
x_plot(m4, xvar = 'simil_jaccard',ylab = 'Likelihood of Match',
       xlab = "Jaccard Similarity",file = "jaccard plot.png")
x_plot(m5, xvar = 'simil_dice', ylab = 'Likelihood of Match',
       xlab = "Dice Similarity",file = "dice plot.png")

# END ##################################################################################
```

B. CHAPTER III CODE

```
##################################################################################
# Title     : Chapter III - TF_IDF Matrix
# Created by: George W. Bailey
# Created on: 4/9/2020
# Some portions of code derived from public sources.
##################################################################################

##################################################################################
# This code will produce a TF-IDF matrix used as text summaries or
# "Topic Models."  This creates topic models as defined in the thesis.
# The code is broken down into sections by "----" and major sections by "####"
##################################################################################


# See reference list for library citations.
library(quanteda)#analysis
library(gtools) # used for mixed sort
library(tidyverse)# used for map_df()
library(textstem) # used for stemming and lemmatisation

##################################################################################
#-------------------------------Read Article Files------------------------------
# Note: uses standard text files as input.

setwd("c:/ADD_YOUR_DIRECTORY/") # You must add your directory!!!!!!!!
```

```
# read all articles
article_text <- list.files(pattern = ".txt")
article_text <- mixedsort(article_text) #sort the files sequentially

# create a dataframe with all of the files and add a new column with file names
# MagicMerge must be unloaded
article_text.df <- map_df(article_text, ~ data_frame(text = read_file(.x)),
                    mutate(fileName = basename(.x)))


#------------------------------------------------------------------------------
# create a corpus from the unstructured text
article_corpus <- corpus(article_text.df)

# stem and lemmatize corpus
article_corpus <- stem_strings(article_corpus)
article_corpus <- lemmatize_strings(article_corpus)

# create tokens from corpus and remove any URLs
article_tokens <- quanteda::tokens(article_corpus, remove_url = TRUE)

#------------------------------------------------------------------------------
# convert tokens to document frame matrix & remove stopwords
article.dfm <- dfm(article_tokens, remove = stopwords('en'))

# convert term frequency measures to to TF_IDF measures
article.dfm <- dfm_tfidf(article.dfm, scheme_tf = "count")
#------------------------------------------------------------------------------
# convert DFM to dataframe (Matrix)
article.df <- convert(article.dfm, to = "data.frame")

# The file will be written in current working directory as "articles_tfidf.csv"
write.csv(article.dfm,"articles_tfidf.csv", row.names = TRUE)

# END ##########################################################################
```

C. CHAPTER IV CODE

```
################################################################################
# Title     : Chapter IV - Weighted Adjacency Matrix (Cosine Similarity)
# Created by: George W. Bailey
# Created on: 4/9/2020
# Some portions of code derived from public sources.
################################################################################

################################################################################
# This code will reproduce the weighted adjacency matrix used by ORA
# in Chapter IV
# The code is broken down into sections by "----" and major sections by "####"
################################################################################


# See reference list for library citations.
```

```
library(quanteda)#analysis
library(gtools) #mixed sort
library(tidyverse)#map_df()
library(textstem) #stemming and lemmatisation

################################################################################
#-------------------------------Read Article Files------------------------------
# Note: uses standard text files as input.

setwd("c:/ADD_YOUR_DIRECTORY/") # You must add you directory!!!!!!!!

# read all articles
article_text <- list.files(pattern = ".txt")
article_text <- mixedsort(article_text) #sort the files sequentially

# create a dataframe with all of the files and add a new column with file names
# MagicMerge must be unloaded
article_text.df <- map_df(article_text, ~ data_frame(text = read_file(.x)),
            mutate(fileName = basename(.x)))


#------------------------------------------------------------------------------
# create a corpus from the unstructured text
article_corpus <- corpus(article_text.df)

# stem and lemmatize corpus
article_corpus <- stem_strings(article_corpus)
article_corpus <- lemmatize_strings(article_corpus)

# create tokens from corpus and remove any URLs
article_tokens <- quanteda::tokens(article_corpus, remove_url = TRUE)

# convert tokens to document frame matrix & remove stopwords
article.dfm <- dfm(article_tokens, remove = stopwords('en'))

# convert term frequency measures to to TF_IDF measures
article.dfm <- dfm_tfidf(article.dfm)


################################################################################
#-------------------Calculate Weighted Adjacency Matrix------------------------
# Note using 'cosine' in the line below.  See Quanteda documentation, or
# Chapter II code for other measures such as 'Dice,' 'Jaccard,' etc...
adj.mat <- as.matrix(as.dist(textstat_simil(article.dfm, method = 'cosine')))

# Output as 'similarity.csv' for input into graphing software
write.csv(adj.mat,"similarity.csv", row.names = TRUE)
#END##############################################################################
```

LIST OF REFERENCES

Banoit, Kenneth et al. *Quanteda: An R Package for the Quantitative Analysis of Textual Data.* (version 2.0.1), 2018. https://quanteda.io.

Beel, Joeran, Bela Gipp, Stefan Langer, and Corinna Breitinger. “Research-Paper Recommender Systems: A Literature Survey.” *International Journal on Digital Libraries*, July 26, 2015, 1–34.

BenarNews. “China’s Military Set to Increase Presence in South China Sea: Pentagon,” September 2, 2020. https://www.benarnews.org/english/news/philippine/Ch-SCS-09022020182723.html.

Carley, Kathleen M. *ORA-LITE* (version 3.0.9.9.116). Windows 64-bit. Pittsburg, PA: Carnegie Mellon University, 2020. http://www.casos.cs.cmu.edu/projects/ora/.

Channel NewsAsia. “Vietnam Says China Military Drills Could Harm Maritime Code Talks,” October 1, 2020. https://www.channelnewsasia.com/news/asia/vietnam-beijing-south-china-sea-military-drills-13169338.

Chen, Minmin. “Efficient Vector Representation for Documents through Corruption.” *ArXiv:1707.02377 [Cs]*, July 7, 2017. http://arxiv.org/abs/1707.02377.

Choi, Seung-Seok, Sung-Hyuk Cha, and Charles Tappert. “A Survey of Binary Similarity and Distance Measures.” *Journal of Systemics, Cybernetics and Informatics* 8, no. 1 (March 10, 2011): 43–48.

Christian, Hans, Mikhael Agus, and Derwin Suhartono. “Single Document Automatic Text Summarization Using Term Frequency-Inverse Document Frequency (TF-IDF).” *ComTech: Computer, Mathematics and Engineering Applications* 7 (December 31, 2016): 285.

Codingest. “China’s Build-up in the South China Sea Is Impacting Commercial Shipping.” Mimicnews, September 28, 2020. https://mimicnews.com/chinas-build-up-in-the-south-china-sea-is-impacting-commercial-shipping.

Cunningham, Daniel, Sean Everton, and Philip Murphy. *Understanding Dark Networks: A Strategic Framework for the Use of Social Network Analysis*. Reprint edition. Lanham: Rowman & Littlefield Publishers, 2016.

Encyclopedia of Mathematics. “Logistic Regression,” March 12, 2016. https://encyclopediaofmath.org/wiki/Logistic_regression.

Freeman, Linton C. “A Set of Centrality Based on Betweenness.” *Sociometry* 40, no. 1 (1977): 35–41.

Gandomi, Amir, and Murtaza Haider. “Beyond the Hype: Big Data Concepts, Methods, and Analytics.” *International Journal of Information Management* 35, no. 2 (April 1, 2015): 137–44. http://www.sciencedirect.com/science/article/pii/S0268401214001066.

Gomaa, Wael H., and Aly A. Fahmy. “A Survey of Text Similarity Approaches.” *International Journal of Computer Applications* 68, no. 13 (April 18, 2013): 13–18. https://doi.org/10.5120/11638-7118.

Greene, D., and P. Cunningham. “Practical Solutions to the Problem of Diagonal Dominance in Kernel Document Clustering.” In *ICML2006*, 377–84. New York: ACM Press, 2006.

Hlavac, Marek. *Stargazer: Well-Formatted Regression and Summary Statistics Tables.* (version 5.2.1), 2018. https://CRAN.R-project.org/package=stargazer.

Huang, Anna. “Similarity Measures for Text Document Clustering.” *Proceedings of the 6th New Zealand Computer Science Research Student Conference*, January 1, 2008.

“IAF Rafales Flying in Ladakh; India-China Military Talks Today.” *Hindustan Times*. 21sep 2020.

“India v China: New Delhi Risks Explosive Backlash in New Move That Will Infuriate Beijing.” *Daily Express*. September 29, 2020. https://www.express.co.uk/news/world/1341387/india-china-war-world-war-3-news-Ladakh-border-china-india-attack.

Kapila, Subhash. “China and India in an Unprecedented ‘State of War’ in September 2020.” *Northlines* (blog), September 28, 2020. http://www.thenorthlines.com/china-and-india-in-an-unprecedented-state-of-war-in-september-2020/.

Laskar, Rezaul H. “China Doubled Its Air Bases, Air Defences and Heliports near LAC in Three Years: Report.” *Hindustan Times*, September 22, 2020. https://www.hindustantimes.com/india-news/china-doubled-its-air-bases-air-defences-and-heliports-near-lac-in-three-years-report/story-LgpbHDWgdfuh4cHwgyeQXP.html.

Manning, Christopher D., and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. 1st ed. Cambridge, Mass: The MIT Press, 1999.

Merriam-Webster. “Definition of Cardinality,” 2020. https://www.merriam-webster.com/dictionary/cardinality.

———. “Definition of Taxonomy,” 2020. https://www.merriam-webster.com/dictionary/taxonomy.

Osborne, Simon. "India v China: Military Chiefs Meet in Bid to Defuse Boiling Cross-Border Tensions." *Daily Express*. September 21, 2020. https://www.express.co.uk/news/world/1338263/india-china-world-war-3-border-dispute-nuclear-armed.

Pemmaraju, Sriram, and Steven Skiena. *Computational Discrete Mathematics: Combinatorics and Graph Theory with Mathmatica.* New York: Cambridge University Press, 2003. https://www.cambridge.org/.

R Core Team. *R* (version 3.6.3). Vienna, Austria, 2020. https://www.R-project.org/.

Radio Free Asia. "Data Shows Commercial Shipping Avoids Hotspots in South China Sea," September 28, 2020. https://www.rfa.org/english/news/china/southchinasea-shipping-09282020155242.html.

Rinker, Tyler W. *{textstem}: Tools for Stemming and Lemmatizing Text* (version 0.1.4). Buffalo, NY, 2018. http://github.com/trinker/textstem.

Roche, Elizabeth. "India-China Military Talks on LAC Disengagement 'Positive' but 'Inconclusive.'" MINT, September 22, 2020. https://www.livemint.com/news/india/india-china-military-talks-on-lac-disengagement-positive-but-inconclusive-11600782120799.html.

"Seventh Round of India-China Military Talks on Border Row Likely next Week." *Hindustan Times*, October 1, 2020. https://www.hindustantimes.com/india-news/seventh-round-of-india-china-military-talks-on-border-row-likely-next-week/story-1DEyccDJrDufi9B6YTqKXM.html.

Srinivasa-Desikan, Bhargav. *Natural Language Processing and Computational Linguistics: A Practical Guide to Text Analysis with Python, Gensim, SpaCy, and Keras*. 1st ed. Birmingham - Mumbi: Packt Publishing, 2018.

Strehl, Alexander, Joydeep Ghosh, and Raymond Mooney. "Impact of Similarity Measures on Web-Page Clustering." In *AAAI 2000*, 7. Austin: AAAI Press, 2001. https://www.aaai.org/Papers/Workshops/2000/WS-00-01/WS00-01-011.pdf.

"Vietnam Says China Military Drills Could Harm Maritime Code Talks." *National Post*. October 1, 2020. https://nationalpost.com/pmn/news-pmn/vietnam-says-china-military-drills-could-harm-maritime-code-talks.

Warnes, Gregory R., Ben Bolker, and Thomas Lumley. *Gtools: Various R Programming Tools* (version 3.8.2), 2020. https://CRAN.R-project.org/package=gtools.

Wickham, Hadley et al. *Tidyverse: Welcome to the Tidyverse* (version 1.3.0), 2019. https://CRAN.R-project.org/package=tidyverse.

Yerepouni Daily News. "Food Is the Best Vaccine against Chaos"; UN Food Agency WFP Wins Peace Nobel," October 9, 2020. https://www.yerepouni-news.com/2020/10/09/food-is-the-best-vaccine-against-chaos-un-food-agency-wfp-wins-peace-nobel/.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California